Proceedings

THIRTY-SECOND ANNUAL ALLERTON CONFERENCE ON COMMUNICATION, CONTROL, AND COMPUTING

September 28-30, 1994

Allerton House, Monticello, Illinois
Sponsored by the
Coordinated Science Laboratory and the
Department of Electrical and Computer Engineering of the
University of Illinois at Urbana-Champaign

PROCEEDINGS

THIRTY-SECOND ANNUAL ALLERTON CONFERENCE ON COMMUNICATION, CONTROL, AND COMPUTING

W. Kenneth Jenkins
Dilip V. Sarwate
Conference Co-Chairs

Conference held
September 28-30, 1994
Allerton House
Monticello, Illinois

Sponsored by
The Coordinated Science Laboratory
and
The Department of Electrical and Computer Engineering
of the
UNIVERSITY OF ILLINOIS
at
Urbana-Champaign

621.3811
ALS
1994

Acknowledgments

This *Proceedings* contains papers presented at the Thirty-second Annual Allerton Conference on Communication, Control, and Computing which was held at Allerton House, Monticello, Illinois, on September 28, 29, and 30, 1994. As in past years, the Conference was sponsored by the Coordinated Science Laboratory and the Department of Electrical and Computer Engineering of the University of Illinois at Urbana-Champaign. It is a pleasure to express our gratitude to Professor Timothy N. Trick, Head of the Department of Electrical and Computer Engineering, for his generous support and constant encouragement which contributed greatly to the success of the Conference.

Forty-five invited papers and fifty-seven contributed papers were presented at the Conference in twenty sessions. We thank the authors for contributing to the success of the Conference. We are also very grateful to our colleagues at the University of Illinois at Urbana-Champaign: Professors B. Bamieh, T. Başar, T. Ü. Başar, R. E. Blahut, Y. Bresler, B. Hajek, P. R. Kumar, U. Madhow, J. V. Medanic, S. P. Meyn, W. R. Perkins, R. Sharma, M. W. Spong, R. S. Sreenivas, A. Vardy, P. G. Voulgaris, and K. A. Zeger for organizing many of the invited sessions, reviewing the contributed papers, chairing sessions, and helping in many other ways in the smooth running of the Conference. Invited sessions were also organized by Dr. C. D. Frank of Motorola, and Professors B. Ghosh of Washington University, St. Louis, M. B. Pursley of Clemson University, C. Schlegel of the University of Texas at San Antonio, and T. Konstantopolous of the University of Texas at Austin, and we are especially grateful to them for their efforts on behalf of the Conference. Professors J. V. Krogmeier of Purdue University, A. Michel of the University of Notre Dame, J. R. Perkins of Boston University, and K. K. Tzeng of Lehigh University chaired contributed sessions at the Conference, and for this we thank them.

The plenary speaker at the Conference was Professor D. G. Messerschmitt, Chairman of the Department of Electrical Engineering and Computer Sciences at the University of California at Berkeley, who gave a thought-provoking lecture titled "Signal Processing and Networking are Intertwined, But Must Be Kept Separate." Professor J. V. Krogmeier of Purdue University also provided much food for thought in his after-dinner speech titled "The Intelligent Vehicle Highway System – Issues and Opportunities." We are grateful to Professors Messerschmitt and Krogmeier for sharing their thoughts on these topics with us. Their excellent presentations contributed much to the success of the Conference. We are also indebted to the University student group The Girls Next Door for providing musical entertainment at the Conference.

We thank Mrs Beverly Becker and her staff at Allerton House for cheerfully accommodating our Conference and for their constant attention to our needs. It is also a great pleasure to thank all the staff members of the Coordinated Science Laboratory who so ably handled all the practical aspects of organizing the Conference. In particular, we wish to mention Mr. Steven Williams and Mrs. Susan Baxley who handled all the business affairs, and Mr. Dan Jordan who handled the sales and shipping of these *Proceedings*. Most of all, however, we want to thank our Conference Secretary, Ms Lila Rhoades, who, with the able assistance of Ms Francie Bridges, Mrs Becky Lonberger, and Ms Dixie Murphy worked all year round in handling announcements, calls for papers, program preparation, Allerton House arrangements, registration, and the final editing of these *Proceedings*. The lion's share of the credit for the success of the Conference belongs to these ladies and it gives us the greatest pleasure to acknowledge this.

W. Kenneth Jenkins

Dilip V. Sarwate

TABLE OF CONTENTS

ACKNOWLEDGMENTS i

I-A: LINEAR CONTROL SYSTEMS

Chairman: B. Bamieh
University of Illinois at Urbana-Champaign

CONTROLLING LPVs WITH INDUCED L_2-NORM PERFORMANCE AND BOUNDS ON PARAMETER VARIATION RATES 1
F. Wu, X. H. Yang, A. Packard and G. Becker

DISCRETE-TIME SYSTEMS: NEW RESULTS ON UNDERSHOOT AND OVERSHOOT 11
L. H. McWilliams and M. K. Sain

A NONRECURSIVE COEFFICIENT ASSIGNMENT AND STABILIZATION SCHEME FOR LINEAR TIME-VARYING SYSTEMS 21
B. K. Ghosh and P. R. Bouthellier

ANALYSIS AND DESIGN OF INTEGRAL CONTROLLERS FOR DISCRETE SYSTEMS 31
J. G. Kuschewski, S. Hui and S. H. Zak

ON THE REACHABILITY AND OBSERVABILITY OF HYBRID MULTIRATE SAMPLING LINEAR SYSTEMS 41
M. de la Sen

ROBUST STABILITY OF A POLYTOPE OF MATRICES: A GLOBAL LINEAR PROGRAMMING SOLUTION 51
B. R. Barmish, C. A. Floudas, C. V. Hollot and R. Tempo

I-B: CODING THEORY I

Organizer and Chairman: A. Vardy
University of Illinois at Urbana-Champaign

THE PERMUTATION TRELLIS COMPLEXITY OF LINEAR BLOCK CODES 60
S. Dolinar, L. Ekroot, A. Kiely, W. Lin and R. J. McEliece

A HEURISTIC FOR ORDERING A LINEAR BLOCK CODE TO MINIMIZE TRELLIS STATE COMPLEXITY 75
F. R. Kschischang and G. B. Horn

ERROR-AND-ERASURE DECODING OF GOLAY CODES VIA CODE DECOMPOSITION 85
B.-Z. Shen and K. K. Tzeng

EFFICIENT SUBOPTIMAL DECODING OF LINEAR BLOCK CODES 93
Y. S. Han, C.-T. Chiu, C. R. P. Hartmann and C. K. Mohan

I-C: COMMUNICATION SYSTEMS

Chair: U. Madhow
University of Illinois at Urbana-Champaign

FREQUENCY OFFSET COMPENSATION IN FREQUENCY FLAT FADING 103
W. Kuo and M. P. Fitz

POWER-BANDWIDTH PROPERTIES OF SHAPED BPSK 113
D. F. De Boer and R. E. Ziemer

OPTIMAL DATA DETECTION OF MPSK IN THE PRESENCE OF UNKNOWN CARRIER PHASE AT LOW COMPLEXITY 121
C. R. Nassar and M. R. Soleymani

A MODIFIED DISTANCE FOR CPM THAT SIMPLIFIES THE SEARCH FOR EXTERNAL CODES 124
Q. Li and B. Rimoldi

AN UPPER BOUND ON SYNCHRONIZATION PROBABILITY FOR MPPM SEQUENCES 131
R. Velidi and C. N. Georghiades

OPTIMAL OPTICAL-DOMAIN FILTERING FOR FIBER-OPTIC COMMUNICATIONS 133
M. Brandt-Pearce

MOBILE CELLULAR CDMA COMMUNICATIONS WITH POWER CONTROL AND SELECTIVE FADING 143
J. H. Gass, Jr., D. L. Noneaker and M. B. Pursley

I-D: STOCHASTIC SYSTEMS

Chairman: J. R. Perkins
Boston University

PARAMETRIZATION OF ALL MEAN-SQUARE STABLE CONTROLLERS AND OBSERVERS FOR DISCRETE-TIME STOCHASTIC HYBRID SYSTEMS 153
E. E. Yaz

STOCHASTIC APPROXIMATIONS FOR STATE-INDEPENDENT DENUMERABLE MARKOV CHAINS 160
Y. Choo and A. Arapostathis

NECESSITY OF KUSHNER-CLARK CONDITION FOR CONVERGENCE OF STOCHASTIC APPROXIMATION ALGORITHMS 167
I.-J. Wang, E. K. P. Chong and S. R. Kulkarni

AN ALGORITHM FOR TRACKING AND DATA ASSOCIATION BY SINGULARITY ESTIMATION 176
A. Satish, R. L. Kashyap and S. Malhotra

ON THE ADAPTIVE DYNAMICAL PWM FEEDBACK REGULATION OF SWITCH-MODE POWER SUPPLIES 186
H. Sira-Ramirez and M. Garcia-Esteban

II-A: CODING THEORY II

Organizer and Chairman: A. Vardy
University of Illinois at Urbana-Champaign

ALTERNATIVES TO COMPLEX CONVOLUTIONAL CODES 196
L. C. Perez and D. J. Costello, Jr.

TWO-DIMENSIONAL MODULATION CODES FOR HOLOGRAPHIC STORAGE 200
A. Vardy, M. Blaum, P. H. Siegel and G. T. Sincerbox

HOMOGENEOUS TRELLIS CODES 210
M. D. Trott and J. P. Sarvis

IMPROVED BINARY CODES AND SEQUENCE FAMILIES FROM Z_4-LINEAR CODES 220
A. G. Shanbhag, P. V. Kumar and T. Helleseth

ON THE CARDINALITY OF THE GROUP OF PERMUTATIONS THAT COMMUTE WITH THE DISCRETE FOURIER TRANSFORM 230
R. Urbanke, Q. Li and B. Rimoldi

II-B: ADVANCES IN COMMUNICATION NETWORKS

Organizer and Chairman: T. Konstantopoulos
University of Texas

OPTIMALITY AND INTERCHANGEABILITY OF LEAKY BUCKETS 235
T. Konstantopoulos and V. Anantharam

ARCHITECTURES FOR LIGHTWEIGHT BUFFERED PACKET SWITCHES 245
R. L. Cruz and J. T. Tsai

BANDWIDTH ALLOCATION FOR MULTIPLE QUALITIES OF SERVICE USING GENERALIZED PROCESSOR SHARING 254
G. De Veciana and G. Kesidis

QUICK DETECTION OF CHANGES IN TRAFFIC STATISTICS: APPLICATION TO VARIABLE RATE COMPRESSION 264
I. Hsu and J. Walrand

THROUGHPUT-DELAY TRADEOFF WITH IMPATIENT ARRIVALS 274
R. Levi and A. Shwartz

II-C: NEURAL NETWORKS AND LEARNING CONTROL

Chairman: A. Michel
University of Notre Dame

STABILITY ANALYSIS OF HOPFIELD NEURAL NETWORKS WITH DELAYS 284
H. Ye, A. N. Michel and K. Wang

SUPERVISED LEARNING IN NEURAL NETWORKS WITHOUT EXPLICIT ERROR BACK-PROPAGATION 294
R. D. Brandt and F. Lin

A DYNAMIC MODEL FOR WINNER-TAKE-ALL COMPETITION 304
C. Baysoy and L. L. Kinney

A NEW LEARNING CONTROL LAW WITH MONOTONIC DECAY OF THE TRACKING ERROR NORM 314
H. S. Jang and R. W. Longman

EXPERIMENTS IN THE USE OF REPETITIVE CONTROL IN BELT DRIVE SYSTEMS 324
Y. S. Ryu, R. W. Longman, E. J. Solcz and J. de Jong

II-D. MULTIUSER DETECTION AND ESTIMATION

Organizers and Chairmen: K. Zeger
University of Illinois at Urbana-Champaign
C. Schlegel
University of Texas at San Antonio

CHANNEL ESTIMATION AND MULTIUSER DETECTION FOR FREQUENCY-NONSELECTIVE FADING SYNCHRONOUS CDMA CHANNEL 335
H.-Y. Wu and A. Duel-Hallen

FACTORISING THE ASYNCHRONOUS CDMA CHANNEL 345
P. D. Alexander and L. K. Rasmussen

A SIMPLE WAY TO COMPUTE THE MINIMUM DISTANCE IN MULTIUSER CDMA SYSTEMS .. 355
C. Schlegel and L. Wei

SOFT DECISION ADAPTIVE MULTIUSER CDMA DETECTOR FOR ASYNCHRONOUS AWGN CHANNELS 362
F. Viehofer, Z. Siveski and Y. Bar-Ness

CYCLIC DECISION FEEDBACK MULTIUSER SEQUENCE DETECTION 372
E. A. Fain and M. K. Varanasi

MULTIUSER RECEIVERS FOR CDMA COMMUNICATION SYSTEMS USING ATENNA ARRAYS 382
R. K. Madyastha and B. Aazhang

PLENARY LECTURE

SIGNAL PROCESSING AND NETWORKING ARE INTERTWINED, BUT MUST BE KEPT SEPARATE
D. G. Messerschmitt
University of California at Berkeley

III-A: ADVANCES IN COMMUNICATION AND MANUFACTURING NETWORKS

Organizer and Chairman: T. Konstantopoulos
University of Texas

LOW COMPLEXITY RANDOMIZED POLICIES FOR MAXIMUM THROUGHPUT IN SYSTEMS WITH CONCURRENCY CONSTRAINTS 392
L. Tassiulas

THE BACCELLI-FOSS SATURATION RULE FOR STABILITY FOR CONTINUOUS TIME INPUT PRO9CESSES 396
E. Altman

LINEAR SWITCHING CURVE CONTROL OF UNRELIABLE MANUFACTURING SYSTEMS 404
J. R. Perkins and R. Srikant

A CLEARING ROUND-ROBIN-BASED STABILIZATION MECHANISM 414
A.F.P. de C. Humes and C. Humes, Jr.

III-B: MOBILE RADIO COMMUNICATIONS

Organizers: M. B. Pursley
Clemson University
C. D. Frank
Motorola, Inc.
Chairman: M. B. Pursley

PPS^{TM}-2000 PERSONAL COMMUNICATIONS SYSTEM: DESCRIPTION AND PERFORMANCE PREDICTION TECHNIQUES AND RESULTS 424
R. Peterson, K. Cutts and J. Haug

PAM SIGNAL SETS FOR CDMA COMMUNICATIONS 434
H.-C. Chu and J. S. Lehnert

ERROR CONTROL TECHNIQUES FOR VOICE AND DATA COMMUNICATION OVER CELLULAR MOBILE CHANNELS 444
S. B. Wicker

CODING ALTERNATIVES FOR DIRECT-SEQUENCE SPREAD-SPECTRUM MULTIPLE-ACCESS COMMUNICATIONS 454
W. E. Stark and J. S. Lehnert

III-C: CODING THEORY III

Chairman: K. K. Tzeng
Lehigh University

LARGE FAMILIES OF SEQUENCES WITH LOW CORRELATIONS AND LARGE LINEAR SPAN 464
A. Klapper

GENERATING FUNCTION BOUNDS FOR CONCATENATED TRELLIS AND REED-SOLOMON CODING 473
J. S. Sadowsky

RLL CODES BASED ON TRELLIS SHAPING 483
M. Siala and G. Kawas Kaleh

CONVERGENCE BEHAVIOR AND DIFFERENTIAL TRELLIS CODE CONSTRUCTION FOR DECODERS USING MULTIPLE SYMBOL DETECTION 493
K. Mackenthun, Jr.

ON THE MEAN SQUARED ERROR OF LINEAR CODES 503
A. S. Khayrallah

III-D: ADAPTIVE AND ROBUST CONTROL

Chairman: P. Voulgaris
University of Illinois at Urbana-Champaign

ON OUTPUT FEEDBACK IN VARIABLE STRUCTURE CONTROL 513
M. K. H. Fan and W.-J. Wang

OBSERVER SYNTHESIS FOR A CLASS OF BILINEAR SYSTEMS 523
R. Martinez-Guerra

SELF-OPTIMIZING ADAPTIVE POLICIES FOR A CLASS OF PARTIALLY OBSERVED CONTROLLED MARKOV CHAINS 525
Y. Choo and A. Arapostathis

REMARKS ON IDENTIFICATION OF CAUSAL DYNAMICAL SYSTEMS 533
G. K. Heitman

IV-A: ROUTING AND SWITCHING IN NETWORKS

Chairman: B. Hajek
University of Illinois at Urbana-Champaign

UNIVERSAL UPPER BOUNDS ON THE PACKET LOSS PROBABILITY OF STATISTICAL MULTIPLEXING IN HIGH-SPEED NETWORKS 543
E. Varvarigos

FAST BYPASS ALGORITHMS FOR HIGH-SPEED NETWORKS 553
I. Cidon, R. Rom and Y. Shavitt

THE ASYMPTOTICS OF TRAFFIC PROCESSES IN LARGE QUEUEING NETWORKS 563
B. Prabhakar, T. S. Mountford and N. Bambos

SHORTEST PATH MULTIPOINT ROUTING: EXPERIMENTAL STUDIES 573
B. M. Waxman

PERFORMANCE BENEFITS OF MULTI-CHANNEL SWITCHING 583
H. Saidi and P. S. Min

WAVELENGTH ASSIGNMENT IN A PASSIVE OPTICAL STAR NETWORK FOR GRID-TRAFFIC PATTERNS 593
G. Pieris, C. Skawratanond and G. Sasaki

IV-B: ADAPTIVE FILTERING

Chairman: J. V. Krogmeier
Purdue University

A NOVEL ADAPTIVE IIR ACOUSTIC ECHO CANCELLER 603
M. Tahernezhadi, L. Liu and G. Miller

STABILITY OF KALMAN FILTERING ALGORITHMS FOR STOCHASTIC DECONVOLUTION PROBLEM 611
S. B. Gelfand and J. V. Krogmeier

A NOVEL ADAPTIVE NOTCH FILTER FOR HOWLING CANCELLATION 617
M. Tahernezhadi, L. Liu and G. Miller

A LOW-COMPLEXITY POLE-ZERO MMSE EQUALIZER FOR ML RECEIVERS 623
N. Al-Dhahir and J. Cioffi

DESIGN OF EXACT CLASSICAL DIGITAL FILTERS 633
L. D. Paarmann and H.-C. Huang

IV-C: ROBOTICS

Chairman: M. W. Spong
University of Illinois at Urbana-Champaign

ISSUES IN THE ZEROS OF FLEXIBLE ROBOT SYSTEMS 641
T. J. Tarn, C. Guo and A. K. Bejczy

COLLISION AVOIDANCE VIA ON-LINE PLANNING AND CONTROL FOR ROBOT MANIPULATOR IN A NON-STATIONARY ENVIRONMENT 651
Z. Yu, K. Ghosh, T. J. Tarn and C. Guo

ACTION PLANNING AND CONTROL FOR ROBOT PHASE TRANSITION: SINGULAR SYSTEMS APPROACH 661
T. J. Tarn, Y. Wu and N. Xi

PLANNING CONDITIONAL SHORTEST PATHS IN AN UNKNOWN ENVIRONMENT 671
R. J. Szczerba, D. Z. Chen and J. J. Uhran, Jr.

H^{∞} CONTROL OF A SINGLE FLEXIBLE LINK ROBOT MANIPULATOR USING A STATE-SPACE APPROACH 673
S. Dindy-Bolongo and E. Sawan

IV-D: DISCRETE EVENT DYNAMIC SYSTEMS I

Organizer and Chair: R. Sreenivas
University of Illinois at Urbana-Champaign

ON THE CONSISTENCY OF HIGH ORDER DERIVATIVE PERTURBATION ANALYSIS ESTIMATORS FOR THE M/G/1 QUEUE 683
L. Shi

PUSH AND PULL SYSTEMS WITH EXTERNAL DEMANDS 691
M. Zazanis

THE SUPERVISORY CONTROL OF AN EXPERIMENTAL MANUFACTURING CELL 699
B. A. Brandin

SYNTHESIS OF CONTROLLERS WITH COLORED PETRI NETS 709
M. Makungu, M. Barbeau and R. St-Denis

SENSOR ASSIGNMENT FOR INVERTIBILITY OF TIMED PETRI NETS 719
Y. Park and E. K. P. Chong

THE USE OF COUPLING IN SENSITIVITY ANALYSIS 728
L. Dai

V-A: SYSTEMS AND CONTROL PROBLEMS IN MACHINE VISION

Organizers and Chairmen: B. Ghosh
Washington University
R. Sharma
University of Illinois at Urbana-Champaign

STEREO VISION VIA MARKOV MODELS 738
D. Geiger, L. T. Maloney and K. Kumaran

ON THE APPLICATIONS OF MORPHOLOGICAL METHODS FOR DYNAMIC OBJECTIVE RECOGNITION 748
P. Bhattacharya and W. Zhu

DETECTION AND TRACKING OF VEHICLES FROM VIDEO WITH APPLICATIONS TO IVHS 755
W. Y. Kan, J. V. Krogmeier and P. C. Doerschuk

V-B: CONTROL IN ELECTRONICS AND SEMICONDUCTOR MANUFACTURING

Organizers and Chairmen: S. Meyn and P. R. Kumar
University of Illinois at Urbana-Champaign

MULTIVARIABLE CONTROL OF PLASMA ENHANCED CHEMICAL VAPOR DEPOSITION USING REAL-TIME MASS SPECTROSCOPY 765
T. J. Knight, X. Cheng, D. W. Greve, B. H. Krogh and M. A. Gibson

CONTROL OF PHOTORESIST PROPERTIES: A KALMAN FILTER BASED APPROACH 774
E. Palmer, W. Ren, C. Spanos and K. Poolla

REAL-TIME FEEDBACK FOR SIDEWALL PROFILE CONTROL IN REACTIVE ION ETCHING 784
B. Rashap, J. Freudenberg and M. Elta

CONTROL ANALYSIS AND DESIGN FOR CZOCHRALSKI GROWTH OF GaAs 794
D. P. Looze, F. Azad, B. Bernstein and T. Collins

V-C: ESTIMATION AND FILTERING

Chairman: Y. Bresler
University of Illinois at Urbana-Champaign

CRAMER-RAO BOUNDS FOR TWO-DIMENSIONAL DOA ESTIMATION USING UNIFORM CIRCULAR ARRAYS 804
A. Y. J. Chan and J. Litva

PERFORMANCE OF DISCRETE-TIME NONLINEAR FILTERS: A CRAMER-RAO BOUND FOR A MORE GENERAL SYSTEM MODEL 814
P. C. Doerschuk

A FEATURE SELECTION ALGORITHM FOR HUMAN SIGNATURE VERIFICATION 822
Y.-J. Chiu and T. Berger

A ROBUST ORDER STATISTIC ESTIMATOR WITH RAPID ADAPTATION 832
P. M. Clarkson and Q. Fan

ON THE APPLICATION OF THE EM ALGORITHM TO SEQUENCE ESTIMATION FOR DEGRADED CHANNELS 841
C. N. Georghiades and J. C. Han

V-D: DISCRETE EVENT DYNAMIC SYSTEMS II

Organizer and Chairman: R. Sreenivas
University of Illinois at Urbana-Champaign

IMPROVED PATH EVALUATION TECHNIQUE FOR CHARACTERIZING UNCONTROLLABLE REACHABILITY IN CONTROLLED PETRI NETS 851
M. Hiriyannaiah and L. E. Holloway

A NEW APPROACH TO SUPERVISORY CONTROL 861
M. A. Shayman and R. Kumar

WHY EVENT OBSERVATION: OBSERVABILITY REVISITED 871
C. Cao and F. Lin

P(L)C-BASED SUPERVISORY CONTROL OF DISCRETE EVENT SYSTEMS 881
B. A. Brandin

MODELING AND ANALYSIS OF SWITCHED-MODE HYBRID SYSTEMS DRIVEN BY THRESHOLD EVENTS 890
B. H. Krogh and T. Niinomi

COMPUTATION OF SUPREMAL ω-CONTROLLABLE LANGUAGES 900
R. Kumar and V. K. Garg

INDEX OF AUTHORS 902

Controlling LPVs with Induced L_2-Norm Performance and Bounds on Parameter Variation Rates

Fen Wu, Xin Hua Yang, Andy Packard* and Greg Becker
Department of Mechanical Engineering
University of California, Berkeley, CA 94720

Abstract

A linear, finite-dimensional plant, with state-space parameter dependence, is controlled using a parameter-dependent controller. The parameters have known, bounded rates of variation, are known to take on values in a compact set, and are known in real-time. The goal of control is to stabilize the parameter-dependent closed-loop system, and provide disturbance/error attenuation as measured in induced $\mathbf{L}_2$ norm. Our approach uses a parameter-dependent Lyapunov function, and solves the control synthesis problem by reformulating the existence conditions into an semi-infinite dimensional convex optimization. We propose finite dimensional approximations to get sufficient conditions for successful controller design.

1 Introduction

In this paper, we consider an ad-hoc approach to reduce the conservatism in quadratic stability analysis of systems with known bounds on the parameter's rate of variation. Our technique is loosely based on the ideas in [23] and [28]. We apply the analysis technique to the design of parameter-dependent controllers for parameter-dependent systems and obtain generalizations of the results in [5], [3] and [2]. A similar method for analysis of parameter-dependent systems has been addressed independently by [10].

The main motivation for our work lies in [15], [14], [28], [30], [29] and [25]. These discuss linear, parameter-varying systems (LPVs) and their importance in gain-scheduling design. Specifically, [15] studies factorizations and realizations of systems over rings. This class of systems includes linear, time-invariant, parameter-dependent systems. In [14], control of time-invariant parameter-dependent systems is considered. The theorems are concerned with the parameter-dependent stabilization, observation and control, and concentrate on the parameter-dependence that is necessary in the controller's state-space entries. Gain-scheduling for linear, parameter-dependent systems is treated in [28]. They derive sufficient conditions for the existence of a parameter-dependent controller which guarantees stability for classes of time-varying parameters. Theoretical issues associated with controlling nonlinear systems using a gain-scheduling perspective are studied in [30]. Several heuristic rules-of-thumb about gain-scheduling are given theoretical interpretation and justification. In [29], a special class of nonlinear systems called "quasi-LPV" are introduced. These types of nonlinear systems can be made to look like linear, parameter-dependent systems using a global differmorphism. Finally, [25] is a complete overview of the extended linearization approach to gain-scheduling.

*The person whom all comments should send to. **Mailing address:** Department of Mechanical Engineering, University of California, Berkeley, CA 94720, USA. **Email:** pack@erg.berkeley.edu, **Phone:** (510) 643-7959, **Fax:** (510) 642-6163.

Consider the parameter-dependent system (modified from [19])

$$\dot{x}(t) = A(\rho(t))x(t) + Bu(t), \tag{1.1}$$

where

$$A(\rho) = \begin{bmatrix} a_{11} & a_{12} & \cos(\rho) & \sin(\rho) \\ a_{21} & a_{22} & -\sin(\rho) & \cos(\rho) \\ 0 & 0 & -\beta & 0 \\ 0 & 0 & 0 & -\beta \end{bmatrix}, \qquad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \beta & 0 \\ 0 & \beta \end{bmatrix}. \tag{1.2}$$

This represents a 2-input system with identical actuator dynamics, a time-varying coupling matrix, and 2nd order plant dynamics. The parameter-dependent state-feedback, quadratic stabilization problem is: find a continuous function $F(\rho)$ and a matrix $P \in \mathcal{S}^{4\times 4}, P > 0$ such that

$$[A(\rho) + BF(\rho)]^T P + P[A(\rho) + BF(\rho)] < 0$$

for all $\rho \in [-\pi, \pi]$. If such matrices existed, then the parameter-dependent state-feedback law

$$u(t) = F(\rho(t))x(t)$$

would render the closed-loop system exponentially stable for any trajectory $\rho(\cdot)$.

Unfortunately, if the matrix

$$A_{11} := \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

is unstable, then this system is not quadratically stabilizable by parameter-dependent state feedback-control. Hence, all of the methods in [5], [4], [3] and [2] are not applicable. The problem is that there are $\rho(\cdot)$ trajectories which allow the upper $(1,2)$ block of $A(\rho)$ in (1.2) to switch between I_2 and $-I_2$ arbitrarily fast. So, regardless of the bandwidth β of the actuators, the rapidly varying parameter $\rho(t)$ do not allow for parameter-dependent quadratic stabilization.

However, a simple singular perturbation argument suggests that the state-feedback

$$F(\rho) := \left[\begin{bmatrix} \cos(\rho) & -\sin(\rho) \\ \sin(\rho) & \cos(\rho) \end{bmatrix} (-\gamma I_2 - A_{11}) \quad 0_{2\times 2} \right]$$

should work (i.e., exponentially stabilize) for $\rho(\cdot)$ trajectories satisfying $\max_{t\geq 0} |\dot{\rho}(t)| \leq B(\beta)$, where $B(\cdot)$ is some monotonically increasing function of β. In other words, if there is a known rate-bound on $\rho(t)$, then exponentially stabilizing, parameter-dependent state-feedbacks do indeed exist. It is possible to produce a parameter-dependent Lyapunov function which demonstrates this stability.

The purpose of this paper is to develop a useful, though somewhat ad-hoc and computationally intensive approach to exploiting rate-of-variation information about the parameters, using parameter-dependent Lyapunov function.

All proofs are omited from this preliminary version.

2 Induced L_2 Norm Analysis

Let $\mathcal{P} \subset \mathbf{R}^M$ be a closed interval (i.e., compact) and $\{\nu_i\}_{i=1}^M$ are non-negative, then we can define **parameter ν-variation set** as

$$\mathcal{F}_{\mathcal{P}}^{\nu} := \left\{ \rho \in C^1(\mathbf{R}, \mathbf{R}^M) : \ \rho(t) \in \mathcal{P}, \ |\dot{\rho}_i(t)| \leq \nu_i, i = 1, \cdots, M, \ \forall\, t \in \mathbf{R} \right\}$$

where ν stands for the variation bound vector $[\nu_1 \cdots \nu_M]^T$.

Consider the behavior of the LPV system governed by

$$\begin{bmatrix} \dot{x}(t) \\ e(t) \end{bmatrix} = \begin{bmatrix} A(\rho(t)) & B(\rho(t)) \\ C(\rho(t)) & D(\rho(t)) \end{bmatrix} \begin{bmatrix} x(t) \\ d(t) \end{bmatrix} \tag{2.1}$$

where $\rho(t) \in \mathcal{P}$, $A: \mathbf{R}^M \to \mathbf{R}^{n \times n}$ is continuous and similarly for B, C and D. It is possible to bound the induced $\mathbf{L}_2$ norm of this system using a parameter-dependent Lyapunov function.

Lemma 2.1 *Given a compact set $\mathcal{P} \subset \mathbf{R}^M$ and the parameter ν-variation set $\mathcal{F}_{\mathcal{P}}^{\nu}$. If there exists a continuously differentiable function $W: \mathbf{R}^M \to \mathcal{S}^{n \times n}$ such that $W(\rho) > 0$ and*

$$\begin{bmatrix} A^T(\rho)W(\rho) + W(\rho)A(\rho) + \sum_{i=1}^{M} \pm \left(\nu_i \frac{\partial W}{\partial \rho_i}\right) & W(\rho)B(\rho) & C^T(\rho) \\ B^T(\rho)W(\rho) & -I & D^T(\rho) \\ C(\rho) & D(\rho) & -I \end{bmatrix} < 0 \tag{2.2}$$

for all $\rho \in \mathcal{P}$, then there exists a scalar δ with $0 \leq \delta < 1$ such that for any $\rho \in \mathcal{F}_{\mathcal{P}}^{\nu}$, the LTV system governed by (2.1) is exponentially stable, and if $d \in \mathbf{L}_2$ and $x(0) = 0$, then

$$\|e\|_2 \leq \delta \|d\|_2$$

Remark 2.1 *The notation $\sum_{i=1}^{M} \pm (\cdot)$ in (2.2) indicates that every combination of $+(\cdot)$ and $-(\cdot)$ should included in the inequality. This means that the 3×3 "inequality" in (2.2) actually represents 2^M different inequalities which must be checked.*

Remark 2.2 *The inequalities in (2.2) represent convex constraints on the set of continuously differentiable functions mapping $\mathbf{R}^M \to \mathcal{S}^{n \times n}$. In section 4, we will discuss some computational approaches to determine the feasibility of these inequalities.*

As is customary [27], [3], [4], [22], [1], [2], we now use this analysis test to derive conditions for control design.

3 Output-feedback Synthesis

Given a parameter-dependent plant, the **LPV Synthesis γ-Performance/ν-Variation Problem** is to determine if there is a parameter-dependent controller and a parameter-dependent Lyapunov function such that the analysis test described in Lemma 2.1 holds for the closed-loop system. In this section we derive necessary and sufficient conditions for this to be possible.

The generalized plant takes on the usual structure

$$\begin{bmatrix} \dot{x}(t) \\ e_1(t) \\ e_2(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} A(\rho(t)) & B_{11}(\rho(t)) & B_{12}(\rho(t)) & B_2(\rho(t)) \\ C_{11}(\rho(t)) & 0 & 0 & 0 \\ C_{12}(\rho(t)) & 0 & 0 & I \\ C_2(\rho(t)) & 0 & I & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ d_1(t) \\ d_2(t) \\ u(t) \end{bmatrix}, \tag{3.1}$$

where $\rho(t) \in \mathcal{P}$, and $x, \dot{x} \in \mathbf{R}^n$, $d \in \mathbf{R}^{n_d}$, $e \in \mathbf{R}^{n_e}$, $u \in \mathbf{R}^{n_u}$ and $y \in \mathbf{R}^{n_y}$. The matrix valued functions are of appropriate dimensions. Note that for simplicity of derivation,

we have assumed that $D_{11}(\rho) = 0, D_{22}(\rho) = 0$ and $D_{12}(\rho), D_{21}(\rho)$ have been scaled to the standard form.

Suppose a linear, parameter-dependent controller is used in feedback from y to u,

$$\begin{bmatrix} \dot{x}_c(t) \\ u(t) \end{bmatrix} = \begin{bmatrix} A_K(\rho(t)) & B_K(\rho(t)) \\ C_K(\rho(t)) & D_K(\rho(t)) \end{bmatrix} \begin{bmatrix} x_c(t) \\ y(t) \end{bmatrix}. \tag{3.2}$$

Define $x_{\rm clp}^T(t) := \begin{bmatrix} x^T(t) & x_c^T(t) \end{bmatrix}$, $e^T(t) := \begin{bmatrix} e_1^T(t) & e_2^T(t) \end{bmatrix}$ and $d^T(t) := \begin{bmatrix} d_1^T(t) & d_2^T(t) \end{bmatrix}$. The closed-loop system can be written as

$$\begin{bmatrix} \dot{x}_{\rm clp}(t) \\ e(t) \end{bmatrix} = \begin{bmatrix} A_{\rm clp}(\rho(t)) & B_{\rm clp}(\rho(t)) \\ C_{\rm clp}(\rho(t)) & D_{\rm clp}(\rho(t)) \end{bmatrix} \begin{bmatrix} x_{\rm clp}(t) \\ d(t) \end{bmatrix}, \tag{3.3}$$

where

$$\begin{aligned} A_{\rm clp}(\rho) &:= \begin{bmatrix} A(\rho) + B_2(\rho)D_K(\rho)C_2(\rho) & B_2(\rho)C_K(\rho) \\ B_K(\rho)C_2(\rho) & A_K(\rho) \end{bmatrix}, \\ B_{\rm clp}(\rho) &:= \begin{bmatrix} B_{11}(\rho) & B_{12}(\rho) + B_2(\rho)D_K(\rho) \\ 0 & B_K(\rho) \end{bmatrix}, \\ C_{\rm clp}(\rho) &:= \begin{bmatrix} C_{11}(\rho) & 0 \\ C_{12}(\rho) + D_K(\rho)C_2(\rho) & C_K(\rho) \end{bmatrix}, \qquad D_{\rm clp}(\rho) := \begin{bmatrix} 0 & 0 \\ 0 & D_K(\rho) \end{bmatrix}. \end{aligned} \tag{3.4}$$

Definition 3.1 *Given a compact set $\mathcal{P} \subset \mathbf{R}^M$, the parameter ν-variation set $\mathcal{F}_{\mathcal{P}}^{\nu}$, performance level $\gamma > 0$, and the open-loop LPV system in (3.1). The* **LPV Synthesis γ-Performance/ν-Variation Problem** *is solvable if there exist an $m \geq 0$, a continuously differentiable matrix function $W : \mathbf{R}^M \to \mathcal{S}^{(n+m)\times(n+m)}$, and continuous bounded matrix functions $(A_K, B_K, C_K, D_K) : \mathbf{R}^M \to (\mathbf{R}^{m\times m}, \mathbf{R}^{m\times n_y}, \mathbf{R}^{n_u\times m}, \mathbf{R}^{n_u\times n_y})$ such that $W(\rho) > 0$ and*

$$\begin{bmatrix} A_{\rm clp}^T(\rho)W(\rho) + W(\rho)A_{\rm clp}(\rho) + \sum_{i=1}^{M} \pm \left(\nu_i \frac{\partial W}{\partial \rho_i}\right) & W(\rho)B_{\rm clp}(\rho) & \gamma^{-1}C_{\rm clp}^T(\rho) \\ B_{\rm clp}^T(\rho)W(\rho) & -I & \gamma^{-1}D_{\rm clp}^T(\rho) \\ \gamma^{-1}C_{\rm clp}(\rho) & \gamma^{-1}D_{\rm clp}(\rho) & -I \end{bmatrix} < 0 \tag{3.5}$$

for all $\rho \in \mathcal{P}$. Here the matrices $A_{\rm clp}, B_{\rm clp}, C_{\rm clp}$ and $D_{\rm clp}$ are defined in (3.4).

We begin with a lemma that could be used to derive state-feedback synthesis result. In this paper, this lemma is only used to prove the sufficient condition in Theorem 3.1.

Lemma 3.1 *Given a compact set $\mathcal{P} \subset \mathbf{R}^M$, $\gamma > 0$, and define $A_F(\rho) := A(\rho) + B_2(\rho)F(\rho)$, $C_F^T(\rho) := \begin{bmatrix} C_{11}^T(\rho) & C_{12}^T(\rho) + F^T(\rho) \end{bmatrix}$. There exist a continuous function $F(\rho)$ and a continuously differentiable function $S(\rho)$ such that $S(\rho) > 0$ and*

$$\begin{aligned} &A_F^T(\rho)S(\rho) + S(\rho)A_F(\rho) + \sum_{i=1}^{M} \pm \left(\nu_i \frac{\partial S}{\partial \rho_i}\right) + \gamma^{-2}C_F^T(\rho)C_F(\rho) \\ &+ S(\rho)\left[B_{11}(\rho)B_{11}^T(\rho) + B_{12}(\rho)B_{12}^T(\rho)\right]S(\rho) < 0 \end{aligned}$$

for all $\rho \in \mathcal{P}$ if and only if there exists a continuously differentiable matrix function $Y(\rho)$ such that for all $\rho \in \mathcal{P}$, $Y(\rho) > 0$ and

$$\begin{bmatrix} Y(\rho)\hat{A}^T(\rho) + \hat{A}(\rho)Y(\rho) + \sum_{i=1}^{M} \pm \left(\nu_i \frac{\partial Y}{\partial \rho_i}\right) - B_2(\rho)B_2^T(\rho) & Y(\rho)C_{11}^T(\rho) & \gamma^{-1}\left[B_{11}(\rho) \;\; B_{12}(\rho)\right] \\ C_{11}(\rho)Y(\rho) & -I & 0 \\ \gamma^{-1}\begin{bmatrix} B_{11}^T(\rho) \\ B_{12}^T(\rho) \end{bmatrix} & 0 & -I \end{bmatrix} < 0,$$

where $\hat{A}(\rho) := A(\rho) - B_2(\rho)C_{12}(\rho)$.

We now state the main theorem for solving the **LPV Synthesis γ-Performance/ν-Variation Problem.**

Theorem 3.1 *Given a compact set $\mathcal{P} \subset \mathbf{R}^M$, the parameter ν-variation set $\mathcal{F}_{\mathcal{P}}^{\nu}$, performance level $\gamma > 0$, and the open-loop LPV system in (3.1), the* **LPV Synthesis γ-Performance/ν-Variation Problem** *is solvable if and only if there exist continuously differentiable matrix functions $X : \mathbf{R}^M \to \mathcal{S}^{n\times n}$ and $Y : \mathbf{R}^M \to \mathcal{S}^{n\times n}$, such that for all $\rho \in \mathcal{P}$, $X(\rho), Y(\rho) > 0$, and*

$$\begin{bmatrix} Y(\rho)\hat{A}^T(\rho) + \hat{A}(\rho)Y(\rho) + \sum_{i=1}^{M} \pm \left(\nu_i \frac{\partial Y}{\partial \rho_i}\right) - B_2(\rho)B_2^T(\rho) & Y(\rho)C_{11}^T(\rho) & \gamma^{-1}B_1(\rho) \\ C_{11}(\rho)Y(\rho) & -I & 0 \\ \gamma^{-1}B_1^T(\rho) & 0 & -I \end{bmatrix} < 0 \tag{3.6.a}$$

$$\begin{bmatrix} \tilde{A}^T(\rho)X(\rho) + X(\rho)\tilde{A}(\rho) + \sum_{i=1}^{M} \pm \left(\nu_i \frac{\partial X}{\partial \rho_i}\right) - C_2^T(\rho)C_2(\rho) & X(\rho)B_{11}(\rho) & \gamma^{-1}C_1^T(\rho) \\ B_{11}^T(\rho)X(\rho) & -I & 0 \\ \gamma^{-1}C_1(\rho) & 0 & -I \end{bmatrix} < 0 \tag{3.6.b}$$

$$\begin{bmatrix} X(\rho) & \gamma^{-1}I \\ \gamma^{-1}I & Y(\rho) \end{bmatrix} \geq 0, \tag{3.6.c}$$

where

$$\hat{A}(\rho) := A(\rho) - B_2(\rho)C_{12}(\rho), \qquad B_1(\rho) = \left[B_{11}(\rho) \;\; B_{12}(\rho)\right],$$
$$\tilde{A}(\rho) := A(\rho) - B_{12}(\rho)C_2(\rho), \qquad C_1^T(\rho) = \left[C_{11}^T(\rho) \;\; C_{12}^T(\rho)\right].$$

If the conditions are satisfied, then by continuity and compactness, it is possible to perturb X such that the three LMIs (3.6.a-3.6.c) still hold and $(X - \gamma^{-2}Y^{-1}) > 0$ uniformly on $\mathcal{P}$. Define $Q(\rho) := X(\rho) - \gamma^{-2}Y^{-1}(\rho), F(\rho) := -\left[B_2^T(\rho)Y^{-1}(\rho) + C_{12}(\rho)\right], L(\rho) := -\left[X^{-1}(\rho)C_2^T(\rho) + B_{12}(\rho)\right]$ and let

$$H(\rho) := -\left[Y^{-1}(\rho)A_F(\rho) + A_F^T(\rho)Y^{-1}(\rho) + C_F^T(\rho)C_F(\rho) + \gamma^{-2}Y^{-1}(\rho)B_1(\rho)B_1^T(\rho)Y^{-1}(\rho)\right],$$

with $A_F(\rho) := A(\rho) + B_2(\rho)F(\rho)$ and $C_F^T(\rho) := \left[C_{11}^T(\rho) \;\; C_{12}^T(\rho) + F^T(\rho)\right]$. One n-dimensional, strictly proper controller K with the state-space data in (3.2) that solves

the feedback problem is defined as:

$$\begin{array}{rcl} A_K(\rho) & := & A(\rho) + \gamma^{-2}\left[Q(\rho)^{-1}X(\rho)L(\rho)B_{12}^T(\rho) + B_1(\rho)B_1^T(\rho)\right]Y^{-1}(\rho) \\ & & + B_2(\rho)F(\rho) + Q^{-1}(\rho)X(\rho)L(\rho)C_2(\rho) - \gamma^{-2}Q^{-1}(\rho)H(\rho) \\ B_K(\rho) & := & -Q^{-1}(\rho)X(\rho)L(\rho) \\ C_K(\rho) & := & F(\rho) \\ D_K(\rho) & := & 0. \end{array} \tag{3.7}$$

4 Computational Considerations

Several ad-hoc approaches to solve the LMIs in Theorem 3.1 can easily be proposed. For instance, let $\{f_i\}_{i=1}^N$ and $\{g_i\}_{i=1}^N$ be user-defined sets of continuously differentiable functions from $\mathbf{R}$ to $\mathbf{R}$. For any matrices $\{X_i\}_{i=1}^N$, $X_i \in \mathcal{S}^{n\times n}$ and $\{Y_i\}_{i=1}^N$, $Y_i \in \mathcal{S}^{n\times n}$, the functions

$$X(\rho) := \sum_{i=1}^N f_i(\rho)X_i, \qquad Y(\rho) := \sum_{i=1}^N g_i(\rho)Y_i$$

are continuously differentiable on $\mathbf{R}^M \to \mathcal{S}^{n\times n}$. Trivially

$$\frac{\partial X}{\partial \rho_j} = \sum_{i=1}^N \frac{\partial f_i}{\partial \rho_j}X_i, \qquad \frac{\partial Y}{\partial \rho_j} = \sum_{i=1}^N \frac{\partial g_i}{\partial \rho_j}Y_i.$$

So, once the basis functions f_i and g_i are chosen, we can attempt to solve the synthesis equations (3.6.) by optimizing over the matrices $X_i, Y_i \in \mathcal{S}^{n\times n}$. The conditions become

$$\begin{bmatrix} \sum_{i=1}^N g_i(\rho)\left(Y_i\hat{A}^T(\rho) + \hat{A}(\rho)Y_i\right) + \sum_{j=1}^M \pm\left(\nu_j \sum_{i=1}^N \frac{\partial g_i}{\partial \rho_j}Y_i\right) & \sum_{i=1}^N g_i(\rho)Y_iC_{11}^T(\rho) & \gamma^{-1}B_1(\rho) \\ -B_2(\rho)B_2^T(\rho) & & \\ C_{11}(\rho)\sum_{i=1}^N g_i(\rho)Y_i & -I & 0 \\ \gamma^{-1}B_1^T(\rho) & 0 & -I \end{bmatrix} < 0$$

$$\begin{bmatrix} \sum_{i=1}^N f_i(\rho)\left(\tilde{A}^T(\rho)X_i + X_i\tilde{A}(\rho)\right) + \sum_{j=1}^M \pm\left(\nu_j \sum_{i=1}^N \frac{\partial f_i}{\partial \rho_j}X_i\right) & \sum_{i=1}^N f_i(\rho)X_iB_{11}(\rho) & \gamma^{-1}C_1^T(\rho) \\ -C_2^T(\rho)C_2(\rho) & & \\ B_{11}^T(\rho)\sum_{i=1}^N f_i(\rho)X_i & -I & 0 \\ \gamma^{-1}C_1(\rho) & 0 & -I \end{bmatrix} < 0$$

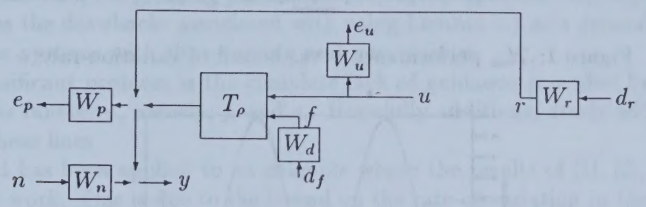
$$\begin{bmatrix} \sum_{i=1}^N f_i(\rho)X_i & \gamma^{-1}I \\ \gamma^{-1}I & \sum_{i=1}^N g_i(\rho)Y_i \end{bmatrix} \geq 0.$$

with $\hat{A}, \tilde{A}, B_1, C_1$ defined as in Theorem 3.1. This represents $2^{M+1}+1$ Affine Matrix Inequalities in the variables X_i, Y_i that must be satisfied for all $\rho \in \mathcal{P}$. Since $\rho \subset \mathbf{R}^M$, it will require approximately L^M points to grid $\mathcal{P}$ with approximately L points in each dimension. So the optimization problem to determine appropriate X_i and Y_i is approximately $L^M\left(2^{M+1}+1\right)$ affine matrix inequalities in the matrix variables $(X_1, Y_1, \ldots, X_N, Y_N)$. The feasibility of these finite number of inequalities can then be determined with several techniques [21], [20], [6], [8], [12]. Theory from semi-infinite optimization should ultimately yield estimates on the grid density that is necessary.

5 Example

As a simple example, we return to the problem which motivated the rate-dependent stabilization. We will only consider output-feedback performance of this problem.

The control problem involves stabilization, tracking, disturbance rejection and input penalty. A block diagram of the generalized plant is shown below.



The plant T_ρ is governed by

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \\ y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} 0.75 & 2 & \cos(\rho(t)) & \sin(\rho(t)) & 0 & 0 & 0 \\ 0 & 0.5 & -\sin(\rho(t)) & \cos(\rho(t)) & 3 & 0 & 0 \\ 0 & 0 & -10 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & -10 & 0 & 0 & 10 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \\ f(t) \\ u_1(t) \\ u_2(t) \end{bmatrix}.$$

The weighting functions are as follows

$$W_p = I_2, \quad W_n = \frac{10(s+10)}{s+1000} I_2, \quad W_d = 1, \quad W_u = \frac{1}{280} I_2, \quad W_r = \frac{20}{s+0.2} I_2.$$

These **roughly** imply objectives of decoupled command response and disturbance rejection, with tracking errors less than 1% of command, and control signals bounded by 280 times the size of $y(t)$.

For this example, we pick $N = 3$, and the basis functions are

$$f_1(\rho) := 1, \quad f_2(\rho) := \cos(\rho), \quad f_3(\rho) := \sin(\rho).$$

These seem natural, given the dependence of the plant on ρ. We choose a rather coarse gridding of interval $\mathcal{P} := [-\pi, \pi]$, namely 6 points, uniformly spaced (i.e., every 60°). First, using the optimal output feedback synthesis [7], we determined that for fixed values of ρ, the optimal achievable γ level is approximately 0.80, and is independent of ρ (although the optimal controllers are strongly dependent on ρ). Using LMIlab [11] with the 6-point gridding, we solved the **LPV Synthesis γ-Performance/ν-Variation Problem** at various levels of ν. The relationship between the optimally achievable γ and ν are shown in the Figure 1.

Since the optimization only guarantees that the AMI holds at the grid points, we further analyzed the resulting solution at 360 points (every 1°). A plot of the negative value of maximum eigenvalue of equation (3.6.c) and equations (3.6.a - 3.6.b) is shown in Figure 2 (the $\nu = 5$ case is displayed), which clearly indicates that the LMIs are indeed solved over the whole interval $\mathcal{P}$.

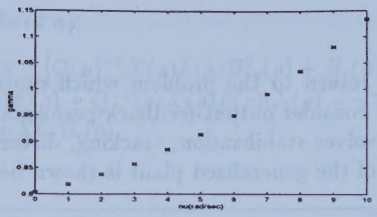


Figure 1: $\mathcal{H}_\infty$ performance γ vs. bound of variation-rate ν

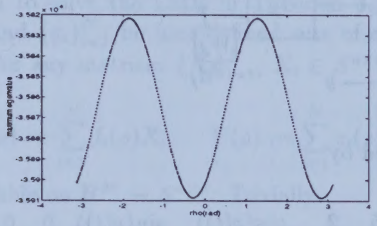


Figure 2: Maximum eigenvalue of LMIs at denser gridding points

A time-varying $\rho(\cdot)$ is also simulated. The trajectory considered is

$$\rho(t) = \pi \sin(\frac{5}{\pi}\, t). \tag{5.1}$$

Note that for the trajectory in (5.1), $\rho(t) \in [-\pi,\ \pi]$ and $|\dot{\rho}(t)| \leq 5$. The simulation results for $d_r(t) = \begin{bmatrix} 0.01 \\ 0 \end{bmatrix}$, $d_f(t), n(t)$ as uniformly distributed random vector in $[-0.01, 0.01]$ are shown in Figure 3.

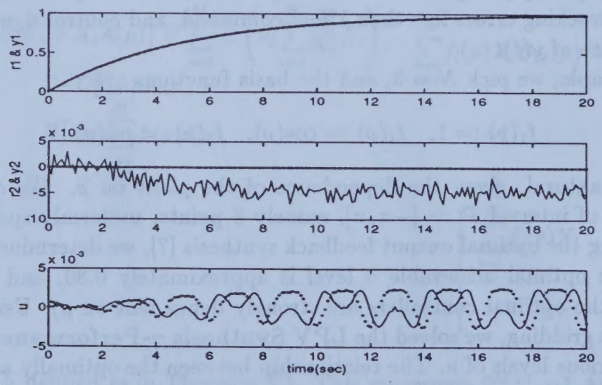


Figure 3: Tracking, error and input responses for time-varying $\rho(t) = \pi \sin(\frac{5}{\pi}\, t)$ with output-feedback controller

6 Comments and Conclusions

Many people will complain about the need to grid the $\mathcal{P}$ set. We too feel that this is a disadvantage of the method. However in many gain-scheduling applications, the number of scheduling variables is small, usually 3 or less. Hence the dimensionality of the gridding, while extremely cumbersome, is not overwhelming. Of course, for a problem with many parameters, the gridding procedure will become prohibitively expensive. This clearly indicates the drawbacks associated with using Lemma 2.1 as a general robustness analysis tool for systems with time-varying real uncertainty.

Another significant problem is the complete lack of guidance provided by the theory to pick the basis functions, namely, f_i and g_i. Hopefully additional study will yield some results along these lines.

This method has been applied to an example where the results of [1], [2], [3], [5], [22] and [5] will not work. This is due to the bound on the rate-of-variation in the parameter. This is pointed out in [13], where a difficult $\mathcal{H}_\infty$ scheduled control law is designed for vertical short take-off and landing vehicles is performed. It would be interesting to try to apply the results here to that specific example. Also note that in Theorem 3.1, by restricting X, Y be constant, we recover the results of [3], [5] and [2], where the quadratic performance problem is considered.

Acknowledgements Independent work with similar objectives is in progress at other institutions [10]. We gratefully acknowledge helpful discussions with G. Balas, S. Boyd, J. Doyle and S. Shahruz, and financial support from the National Science Foundation, awards ECS-9096223 and CTS-9057420.

References

[1] P. Apkarian and P. Gahinet, "A Convex Characterization of Parameter Dependent $\mathcal{H}_\infty$ Controllers", to appear on *IEEE Trans. Automatic Control.*

[2] P. Apkarian, P. Gahinet and G. Becker, "Self-scheduled $\mathcal{H}_\infty$ Control of Linear Parameter Varying Systems, in *Proc. American Control Conf.*, 1994, pp. 856-860.

[3] G. Becker, *Quadratic Stability and Performance of Linear Parameter Dependent Systems*, Ph.D. dissertation, Department of Mechanical Engineering, University of California at Berkeley, CA, Dec. 1993.

[4] G. Becker and A. Packard, "Robust Performance of Linear Parametrically Varying Systems Using Parametrically Dependent Linear Dynamic Feedback", to appear on *Systems and Control Letters.*

[5] G. Becker, A. Packard, D. Philbrick and G. Balas, "Control of Parametrically-Dependent Linear Systems: A Single Quadratic Lyapunov Approach", in *Proc. American Control Conf.*, 1993, pp. 2795-2799.

[6] B. Boyd and L. El Ghaoui, "Method of Centers for Minimizing Generalized Eigenvalues", *Linear Algebra and its Applications*, vol. 188, pp. 63-111, 1992.

[7] J.C. Doyle, K. Glover, P.P. Khargonekar and B. Francis, "State-space Solutions to Standard $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control problems", *IEEE Trans. Automatic Control*, vol. 34, pp. 831-847, 1989.

[8] M. Fan and B. Nekooie, "A Second Order Interior Point Method for Solving Linear Matrix Inequality Problems", in *Proc. American Control Conf.*, 1994, pp. 831-835.

[9] P. Gahinet, "A Convex Parametrization of $\mathcal{H}_\infty$ Suboptimal Controllers", in *Proc. 31st IEEE Conf. Decision and Control*, 1992, pp. 937-942.

[10] P. Gahinet, *personal communication*, June 1994.

[11] P. Gahinet and A. Nemirovskii, LMI lab: A Package for Manipulating and Solving LMIs (Ver. 2.0), 1993.

[12] J. Haeberly and M. Overton, "Optimizing Eigenvalues of Symmetric Definite Pencils", in *Proc. American Control Conf.*, 1994, pp. 836-839.

[13] R. Hyde and K. Glover, "The Application of Scheduled $\mathcal{H}_\infty$ Controllers to A VSTOL Aircraft", *IEEE Trans. Automatic Control*, vol. 38, pp. 1021-1039, 1993.

[14] E. Kamen and P.P. Khargonekar, "On the Control of Linear Systems Whose Coefficients are Functions of Parameters", *IEEE Trans. Automatic Control*, vol. 29, pp. 25-33, 1984.

[15] P.P. Khargonekar and E. Sontag, "On the Relation Between Stable Matrix Fraction Factorizations and Regulable Realizations of Linear Systems Over Rings, *IEEE Trans. Automatic Control*, vol. 27, pp. 627-638, 1982.

[16] W. Lu, K. Zhou and J.C. Doyle, "Stabilization of LFT systems", in *Proc. 30th IEEE Conf. Decision and Control*, 1991, pp. 1239-1244.

[17] W. Lu and J. Doyle, "$\mathcal{H}_\infty$ Control of LFT systems: An LMI Approach", in *Proc. 31st IEEE Conf. Decision and Control*, 1992, pp. 1997-2001.

[18] W. Lu and J. Doyle, "$\mathcal{H}_\infty$ Control of Nonlinear Systems: A Convex Characterization", in *Proc. American Control Conf.*, 1994, pp. 2098-2102.

[19] G. Meyer and L. Cicolani, "Application of Nonlinear Systems Inverses to Automatic Flight Control Design–System Concepts and Flight Evaluations", *AGARDograph: Theory and Applications of Optimal Control in Aerospace Systems*, No. 251, 1981.

[20] A. Nemirovskii and P. Gahinet, "The Projective Method for Solving Linear Matrix Inequalities", in *Proc. American Control Conf.*, 1994, pp. 840-844.

[21] Y. Nesterov and A. Nemirovskii, Interior Point Polynomial Methods in Convex Programming: Theory and Applications, Lecture notes in mathematics, pringer Verlag, 1992.

[22] A. Packard, "Gain Scheduling via Linear Fractional Transformations", *Systems and Control Letters*, vol. 22, pp. 79-92, 1994.

[23] A. Packard and G. Becker, "Quadratic Stabilization of Parametrically Dependent Linear Systems Using Parametrically Dependent Linear Dynamic Feedback", *1992 ASME Winter Annual Meeting*, vol. 43, pp. 29-36, 1992.

[24] A. Packard, K. Zhou, P. Pandey and G. Becker, "A Collection of Robust Control Problems Leading to LMIs", in *Proc. 30th IEEE Conf. Decision and Control*, 1991, pp. 1245-1250.

[25] W. Rugh, "Analytical Framework for Gain Scheduling", *IEEE Control Systems Magazine*, vol. 11, pp. 74-84, 1991.

[26] M. Sampei, T. Mita and M. Nakamichi, "An Algebraic Approach to $\mathcal{H}_\infty$ Output Feedback Control Problems", *Systems and Control Letters*, vol. 14, pp. 13-24, 1990.

[27] C. Scherer, "$\mathcal{H}_\infty$ Optimization without Assumptions on Finite or Infinite Zeros", *SIAM J. Control and Optimization*, vol. 30, pp. 143-166, 1992.

[28] S. Shahruz and S. Behtash, "Design of Controllers for Linear Parameter Varying Systems by The Gain Scheduling Technique", *J. Mathematical Analysis and Applications*, vol. 168, pp. 195-217, 1992.

[29] J. Shamma and M. Athans, "Gain Scheduling: Potential Hazards and Possible Remedies", *IEEE Control Systems Magazine*, vol. 12, pp. 101-107, 1992.

[30] J. Shamma and M. Athans, "Analysis of Nonlinear Gain Scheduled Control Systems", *IEEE Trans. Automatic Control*, vol. 35, pp. 898-907, 1990.

DISCRETE-TIME SYSTEMS: NEW RESULTS ON UNDERSHOOT AND OVERSHOOT *

Leo H. McWilliams
AlliedSignal Braking Systems
South Bend, Indiana 46634

Michael K. Sain
Department of Electrical Engineering
University of Notre Dame
Notre Dame, Indiana 46556

ABSTRACT

Digital revolution notwithstanding, virtually no work has been reported on the qualitative response features of discrete-time systems. This paper considers such systems, described by a rational transfer function which is strictly proper, has real coefficients, and satisfies a basic and natural condition concerning the dominant response mode. A new and fundamental result on initial undershoot is established. This result employs a definition of initial undershoot which extends that which is usually used in the literature, and is able to include responses which grow without bound and which decay geometrically to zero. With regard to unit step responses, new results are presented for final overshoot. Lower bounds are proved for undershooting characteristics, insofar as they relate to real zeros, and sufficient conditions are given for these bounds to be achieved.

1. INTRODUCTION

The z-transform technique gained momentum in the 1950's with the publication of several papers which developed the mathematical foundation for analysis and design in the z-domain [1–4]. However, very little seems to have been published on the limitations imposed by zeros on the transient response of the system.

In a series of papers, E. I. Jury examined the relationship between the transient response and the pole-zero pattern of the system. In 1954, Jury studied the step response of a closed loop system with an open loop transfer function consisting of a single pole at $z = 1$ and a positive real pole to the left of this pole [1]. This system has no open loop zeros. Constant overshoot and phase contours were plotted as a function of the positive pole not located at $z = 1$. A relationship between the maximum of the frequency response of the system and the maximum of the time response was developed in this work. In 1955, Jury also developed conditions for monotonic and nonmonotonic time responses. These conditions were expressed in terms of the pole and zero locations [2].

*The work of the second author was supported by the University of Notre Dame's Frank M. Freimann Chair in Electrical Engineering.

He also demonstrated that under appropriate conditions, the transient response may be approximated by a single dominant term. This approximation enabled the extension of synthesis techniques for second order systems to include general pole-zero patterns. Jury's results were similar in nature to Mulligan's results [5] that were developed for the Laplace transform. Jury continued the analysis of the z-transform and its relationship to the time response with the presentation of a correlation between the root locus and transient response of a sampled-data system [3]. In this work, Jury studied the effect of an additional real, positive zero within the unit circle on the step response of a system whose dominant poles are complex. This analysis investigated the change in peak overshoot, M_p, and the time to peak overshoot, n_p. He noted that, as the real zero approaches $z = 1$ from the left, the overshoot increases. In most practical cases, overshoot is undesirable. This overshoot was not evident once the zero shifted to the right past $z = 1$. We will show in this paper that this group of zeros imposes very specific response limitations on the transient response.

G.W. Johnson and D.P. Lindorff also studied the transient response of sampled-data systems [4]. These authors assumed a linear, continuous-time system in series with an ideal sampler arranged in a feedback configuration. They attempted to show through computer simulation that any type-1 system that has not been partially equalized by lag network compensation can be approximated by the pseudo system

$$G(s) = \frac{K}{s(\tau s + 1)^n}.$$

The values of n and τ are design values chosen to provide a close match in the frequency domain between the pseudo system and the actual system.

This paper initiates further work on the qualitative response features of linear, discrete-time systems. It considers systems having a rational transfer function which is strictly proper, has real coefficients, and satisfies a basic and natural condition concerning the dominant response mode. For this class, a fundamental theorem on initial undershoot is established. The definition of initial undershoot is extended beyond that which is usually discussed in the literature, and is able to include responses which grow without bound and which decay geometrically to zero. With regard to unit step responses, new results are presented for final overshoot. Lower bounds are proved for undershooting characteristics, insofar as they relate to real zeros, and sufficient conditions are given for these bounds to be achieved.

2. PRELIMINARIES

Consider a linear, time invariant, single input, single output, strictly causal, finite-dimensional, discrete-time system. The inputs and outputs of the system will be understood as real-valued sequences and will be denoted by $(u(k))$ and $(y(k))$, respectively. It is assumed that these sequences have finite left support, which is to say that they vanish for indices k which are less than some finite integer, that can depend upon the sequence in question. Correspondingly, we denote the system by $(h(k))$, where $h(k) = 0$ when $k \leq 0$. The output sequence corresponding to a given input sequence can then be represented in the classic convolutional form by the relation

$$y(k) = \sum_{l=0}^{k-1} h(k-l)u(l),$$

when the input sequence does not begin until index zero. Of course, in the analysis, there is special interest attached to such input sequences which are zero for negative indices. The z-transform of such a sequence $(u(k))$, where $u(k) = 0$ when k is less than 0, is defined by

$$U(z) = \sum_{k=0}^{\infty} u(k) z^{-k},$$

and is assumed to converge for $|z| > R_0$ [1]. It will be sufficient for the purposes of the following discussion if we further assume that $U(z)$ is a rational transfer function. The z-transform $Y(z)$ of the output sequence $(y(k))$ is then found by the customary relationship $Y(z) = P(z)U(z)$, where $P(z)$ denotes the z-transform of $(h(k))$.

A theory which establishes response limitations imposed by real zeros of $P(z)$ is presented in this paper. The unit step sequence, which is defined as

$$\begin{aligned} u(k) &= 0, \; k < 0 \\ u(k) &= 1, \; k \geq 0 \end{aligned}$$

plays a central role in the results. But, and it is important to bear this in mind, the step response is frequently just an intermediate tool to be employed on the way to a more general result.

We shall be primarily interested in sequences with a finite number of sign changes. Toward this end, we introduce the following definition. The outermost poles of a transfer function are understood to be those whose complex magnitudes are a maximum among all the complex magnitudes of the poles of the function.

Definition: A rational transfer function is said to satisfy Condition D if the set of its outermost finite poles is a singleton, which is real and positive.

3. A BASIC RESULT ON INITIAL UNDERSHOOT

This section gives necessary and sufficient conditions for a response sequence to exhibit initial undershoot. The results are rather general in nature, and hold for systems having transfer functions which satisfy Condition D. It is not, for example, required that the excitation be a step sequence, or even that the response be bounded. In fact, the response sequence may also approach zero asymptotically. Accordingly, the basic theorem has quite broad applicability. The treatment, however, makes use of a pole-shifting technique which brings the transfer functions into a form much like that of a step response.

Let the unit impulse sequence be that sequence which has unit value at index zero and is zero for other indices, and let $P(z)$ be the z-transform of the unit impulse response of the system. Now express $P(z)$ in the manner

$$P(z) = \frac{Cn(z)}{d(z)}$$

with $n(z)$ and $d(z)$ being monic, coprime polynomials. The initial sign of a response $(y(k))$ to an excitation $(u(k))$ is defined as the sign of the first nonzero value of $(y(k))$.

If $P(z)$ has relative degree r, $r \geq 0$, then

$$\lim_{z \to \infty} z^k Y(z) = \begin{matrix} 0, & k < r \\ C, & k = r \end{matrix}$$

which implies that

$$\begin{aligned} y(k) &= 0, \; k < r \\ y(k) &= C, \; k = r. \end{aligned}$$

Thus the initial sign of the unit impulse response sequence is determined to be that of the constant C.

Next, suppose that the system were excited with a unit step sequence. Then the z-transform of the unit step response is

$$Y(z) = \frac{zP(z)}{(z-1)}.$$

If $P(z)$ had no poles with complex magnitudes greater than or equal to unity, then the final value of (y(k)) would be

$$\lim_{z \to 1}(1 - z^{-1})Y(z) = \lim_{z \to 1} P(z) = \frac{Cn(1)}{d(1)} = C\gamma$$

where, for the present, it is assumed that

$$\gamma = \frac{n(1)}{d(1)} \neq 0.$$

This is a most convenient observation for determining the sign of final values. The step response case suggests its generalization.

Suppose that $P(z)$ satisfies Condition D; then the final sign of $(h(k))$ is defined to be the sign of $h(k)$ when k is chosen so large that $h(k)$ undergoes no further sign changes. In order to state precisely what this means, two cases must be considered. For the first case, $P(z)$ has its pole of largest complex magnitude at $z = 1$ in the complex plane. With a modest redefinition of terms, this case is directly analogous, if not identical, to the step response discussion of the paragraph preceding. For the second case, the designated pole of Condition D may lie either inside the unit circle centered at the origin, or outside. If it is inside, the system is called stable; it it is outside, the system is called unstable. Without loss of generality, suppose that it is outside. Then choose $\alpha > 0$ such that

$$(v(k)) = (\alpha^{-k} h(k))$$

is formed. Moreover,

$$\text{sgn}(v(k)) = \text{sgn}(h(k)).$$

The pole-zero pattern of $V(z)$ is a radially shifted version of the pole-zero pattern of $P(z)$. Choose α to shift the rightmost pole of $P(z)$ to $z = 1$. Because the rightmost pole of $P(z)$, hence $V(z)$, is simple, then the final value of $(v(k))$ is

$$\lim_{k \to \infty} v(k) = \lim_{z \to 1}(1 - z^{-1})P(\alpha z),$$

which is nonzero. Hence, the sign of this final value of $(v(k))$ yields the final sign of $(h(k))$. With these ideas in place, it is possible to give the following definition for initial undershoot in the unit impulse response of systems satisfying Condition D.

Definition: The unit impulse response of the system described by $P(z)$, where $P(z)$ is a strictly proper and rational transfer function which satisfies Condition D, displays initial undershoot if its initial sign differs from its final sign. With straightforward modifications, this notion extends to the response of the system described by $P(z)$ to any input sequence $(u(k))$ with proper rational z-transform.

It is now possible to establish a necessary and sufficient condition for initial undershoot.

Theorem 1: Let $F(z)$ be a strictly proper, rational transfer function satisfying Condition D with outermost pole r. The inverse transform of $F(z)$, denoted $(f(k))$, exhibits initial undershoot if and only if $F(z)$ has an odd number of real, positive zeros with magnitude greater than r.

For step responses, Theorem 1 provides the following corollary.

Corollary 1: Let $P(z)$ satisfy the assumptions of Theorem 1. Suppose further that $P(z)$ has no zero at the point $z = 1$, and that r is not equal to 1. If (u(k)) is the unit step sequence, then the unit step response (y(k)) displays initial undershoot if and only if there is an odd number of real zeros to the right of the outermost pole of $Y(z)$. If $P(z)$ is stable, then these zeros are nonminimum phase. If $P(z)$ is unstable, then these zeros are to the right of the outermost pole of $P(z)$.

Remark: $P(z)$ can be stable or unstable, as long as the restriction of Condition D is met. This approach to the meaning of initial undershoot extends the customary meaning to a much wider class of functions.

4. NEW RESULTS ON FINAL OVERSHOOT

When the input sequence is a unit step, the output simplifies to

$$y(k) = \sum_{l=1}^{k} h(l).$$

Each value of $h(l)$ for $1 \leq l \leq k$ has an equal weight in the determination of $y(k)$. In addition, the first nonzero value of $y(k)$ occurs at $k = r$ where r is the relative degree of $P(z)$ and is simply $h(r)$. Notice that $y(0)$ is equal to zero because $h(0)$ is equal to zero. This means that the response always begins at zero. Consider two consecutive time instants l_1 and l_2, with the second being the larger. If

$$\text{sgn}(h(l_2)) = 1,$$

then

$$y(l_1) < y(l_2).$$

Similarly, if

$$\text{sgn}(h(l_2)) = -1,$$

then

$$y(l_1) > y(l_2).$$

Suppose that the final value of $(y(k))$ exists and is positive. Choose l_1 such that $(h(k))$ exhibits no sign changes for $k > l_1$. If $(h(k))$ is negative in this region and C_1 is the steady state value of $y(k)$, then

$$y(k) > C_1, \; k > l_1$$

and $(y(k))$ overshoots its final value in this region. Similarly, if $h(k)$ is positive in this region, then

$$y(k) < C_1, \; k > l_1$$

and $(y(k))$ does not overshoot in this region. Final overshoot is summarized in the definition that follows.

Definition: Suppose that C_1 is the nonzero final value of the step response for a system with impulse response (h(k)). If there exists l_1 such that

$$\text{sgn}(h(k)C_1)) = -1, \; k > l_1$$

then $(y(k))$ exhibits final overshoot.

A necessary and sufficient condition for such a $(y(k))$ to exhibit final overshoot is stated in the theorem that follows.

Theorem 2: Let $P(z)$ be a stable and strictly proper transfer function. Suppose further that $P(z)$ satisfies Condition D, with outermost pole located at p_1, and that $P(1) \neq 0$. The unit step response of the system exhibits final overshoot if and only if $P(z)$ has an odd number of real zeros in the range of z defined by

$$0 < p_1 < z < 1.$$

5. FURTHER RESULTS ON UNDERSHOOT

This section continues the analysis with the presentation of a lower bound on the undershoot exhibited by the unit step response. The number of sign changes exhibited by the response is important. To this end, the following definition will be useful.

Definition: If $(y(k))$, defined for $k \geq 0$, has a finite number q of sign changes there, then there are $q + 1$ instances, denoted k_i,

$$0 \leq k_1 < k_2 < \ldots < k_{q+1}$$

which are not unique, such that

$$\operatorname{sgn}(y(k_i)y(k_{i+1})) = -1;$$

and $(y(k))$ exhibits type-q undershoot.

The next theorem addresses the effect of an added zero on the time domain response of the system.

Theorem 3: Let $V(z)$ be a strictly proper transfer function with relative degree at least two. Further assume that $V(z)$ satisfies Condition D. Let $(v(k))$ be the inverse z-transform of $V(z)$ and assume that $(v(k))$ exhibits type-q undershoot. Consider the function $Y(z)$ formed from $V(z)$ in the manner

$$Y(z) = CV(z)(-\tau z + 1)$$

with C a real, nonzero constant and τ a real positive constant chosen such that the zero introduced at $z = 1/\tau$ is to the right of the rightmost pole of $V(z)$. If $(y(k))$ is the inverse z-transform of $Y(z)$, then $(y(k))$ exhibits at least type-$q+1$ undershoot.

It is clear from Theorem 3 that the critical zeros in the case of undershoot are real, positive and located to the right of the outermost pole. These zeros always exert their influence on the response when considering the undershoot. In addition, a lower bound on the undershoot is available, by simply recursively adding the real zeros to a base response to arrive at the conclusion that q real positive zeros to the right of the outermost pole lead to a response that exhibits at least type-q undershoot.

When $0 < p_1 \neq 1$, then we follow the same adjustment procedure which was used in Theorem 1.

In the case of stable $P(z)$, stronger results for step responses follow.

Theorem 4: Let $H(z)$ be a stable, strictly proper, rational transfer function with $H(1) \neq 0$. The step response of $H(z)$ exhibits initial undershoot if and only if $H(z)$ has an odd number of real, positive zeros outside the unit circle.

Theorem 5: Let $H(z)$ satisfy the assumptions of Theorem 4, and have relative degree at least two. Similarly to Theorem 3, define

$$P(z) = CH(z)(-\tau z + 1),$$

with $0 < \tau < 1$. If the step response of $H(z)$ exhibits type-q undershoot, then the step response of $P(z)$ shows at least type-$(q+1)$ undershoot.

Remark: It should be noted in these two theorems that Condition D is not required.

The next section considers some cases in which bounds can be tight.

6. REACHING THE BOUNDS

This section presents classes of transfer functions for which the bounds presented above can become exact. The results to be presented are based on a theorem of Laguerre [7]. Laguerre's result is an extension of Descartes' Rule of Signs. The presentation begins with a review of this rule. Laguerre's result and its significance are then presented. A lemma which allows Laguerre's theorem to be used in the proofs of the main results can then be stated.

Descartes' Rule of Signs establishes an upper bound on the number of real, positive roots of the polynomial equation

$$f(x) = \sum_{i=0}^{n} B_i x^i = 0.$$

Denote by $\{A_i\}$ the sequence of nonzero elements of $\{B_i\}$. By a sign variation in the coefficient sequence, we mean that

$$\text{sgn}(A_i A_{i+1}) = -1$$

for some value of i. The upper bound on the number of roots is based on the total number of such sign variations.

In addition, Descartes' result determines the possible numbers of real, positive roots when the upper bound is not met.

Descartes' Rule of Signs [8]: The number of real positive roots of the equation

$$f(x) = 0,$$

where $f(x)$ is a real polynomial of degree n, is at most equal to the number of sign variations in the sequence of its nonzero coefficients. If the number of real positive roots is less than this number, then it is always less by a multiple of two.

Laguerre proved a version of this theorem for the equation

$$f(x) = \sum_{i=0}^{n} A_i e^{p_i x} = 0$$

where p_i, $i = 1, \ldots, n$ are real numbers satisfying $p_i < p_i + 1$. Laguerre establishes the validity of the first conclusion from Descartes' theorem.

Laguerre's Theorem [7]: The number of real positive roots of the equation

$$f(x) = \sum_{i=0}^{n} A_i e^{p_i x} = 0$$

is at most equal to the number of sign variations in the sequence of its coefficients.

The upper bound developed by Laguerre is in the same spirit as that developed by Descartes. However, the second part of Descartes' theorem is not valid.

Let $P(z)$ be a strictly proper transfer function. Assume that all the poles of $P(z)$ are real and simple. It is possible to determine $(p(k))$ from a partial fraction expansion of $P(z)$ with minor modifications. If p_i is a pole of $P(z)$, then the partial fraction expansion of $P(z)$ is simply

$$P(z) = \sum_{i=0}^{n} \frac{A_i}{z - p_i}.$$

The individual terms of these z-transforms do not readily invert. Terms of the form $A_i z/(z - p_i)$ have inverse z-transform $(A_i p_i^k)$. Thus, if $P(z)$ has a single zero at the origin, then it is more advantageous to expand $P(z)/z$ into its partial fractions and multiply the result by z to get $P(z)$. The individual terms in the expansion have the desired form allowing straightforward inversion [9]. If $P(z)$ has no zeros at the origin, then the partial fraction expansion of $P(z)$ can be calculated directly. One can then construct

$$P_1(z) = zP(z),$$

and the inverse z-transform of $P_1(z)$ is computed. If $(p_1(k))$ is the impulse response of the system described by $P_1(z)$, then

$$p(k) = p_1(k - 1).$$

The inverse transform in this case has a unit delay.

The inverse transform of $P(z)$ is of the form

$$p(k) = \sum_{i=1}^{n} A_i p_i^j$$

where j and A_i depend on whether or not $P(z)$ has a zero at the origin. If $P(z)$ has a single zero at the origin, then $j = k$. If $P(z)$ does not have a zero at the origin, then $j = k - 1$. If all poles of $P(z)$ are nonnegative, then it is possible to express $p(k)$ as

$$p(k) = \sum_{i=1}^{n} A_i e^{j \ln p_i}, \quad k \geq 0.$$

If j and k are replaced by x and x is assumed to be nonnegative, then $p(x)$ is continuous. In order for $p(x)$ to change signs in a region, then there must exist a point x_1 in this region such that

$$p(x_1) = 0.$$

Clearly, if $p(x)$ undergoes q sign changes, then $p(x)$ must have at least q such zeros. The number of sign changes of $p(x)$ can never be more than the number of its positive roots. This in turn bounds the number of sign changes of $(p(k))$.

The theorem that follows presents a class of transfer functions for which the number of sign changes in the impulse response and the number of real positive zeros to the right of the rightmost pole are equal. A corollary is then presented which further examines the number of sign changes.

Theorem 6: Assume that $P(z)$ is a strictly proper rational transfer function which satisfies Condition D and has no complex poles or complex zeros. Further assume that

$P(z)$ has no negative real poles. If $P(z)$ has precisely m zeros all of which are real and to the right of the rightmost pole of $P(z)$, and at least $m+1$ simple positive poles, then the step and impulse responses have precisely m sign changes.

Remark: In a recent paper [10], El-Khoury, Crisalle, and Longchamp consider the problem of numbers of local extrema in the step responses of stable systems. If one is interested in local extrema, then every sign change produces one; on the other hand, one can see large numbers of local extrema without corresponding sign changes. Because of the importance of undershoot phenomena, it is therefore crucial to study response sign changes in their own right. The present paper introduces techniques applicable to cases which have multiple real poles and which may not be stable. Reference [10] does not present proofs for these very interesting situations. However, in the stable casee, [10] considers some transfer functions not addressed here. Thus, the two works are best viewed as complementary.

REFERENCES

[1] E. I. Jury, "Analysis and synthesis of sampled data control systems," *AIEE Transactions,* vol. 73, pt. I, Sept. 1954, pp. 332–346.

[2] E. I. Jury, "The effect of pole and zero locations on the transient response of sampled data control systems," *AIEE Transactions,* vol. 74, pt. I, March 1955, pp. 41–48.

[3] E. I. Jury, "Correlation between root locus and the transient response of sampled-data control systems," *AIEEE Transactions,* vol. 75, pt. I, January 1956, pp. 427.

[4] G. W. Johnson and D. P. Lindorff, "Transient analysis of sampled-data control systems," *AIEE Transactions,* vol. 73, pt. I, July 1954, pp. 147-153.

[5] J. H. Mulligan, "The effect of pole and zero locations on the transient response of linear dynamical systems," *Proceedings of the IRE,* vol. 37, May 1949, pp. 516–529.

[6] E. I. Jury, *Theory and Applications of the z-Transform Method,* New York, John Wiley and Sons, 1964.

[7] E. Laguerre, *Oeuvres,* vol. 1, p. 29.

[8] J. V. Uspensky, *The Theory of Equations,* New York, McGraw Hill, 1948.

[9] B. Kuo, *Digital Control Systems,* New York, Holt Rhinehart and Winston Inc. 1980.

[10] M. El-Khoury, O. D. Crisalle, and R. Longchamp, "Discrete Transfer-Function Zeros and Step-Response Extrema," *Proceedings Twelfth IFAC World Congress,* vol. 3, 1993, pp. 81-86.

A NONRECURSIVE COEFFICIENT ASSIGNMENT AND STABILIZATION SCHEME FOR LINEAR TIME-VARYING SYSTEMS

B. K. Ghosh
Department of Systems Science and Mathematics
Box 1040
Washington University
St. Louis, MO 63130

P. R. Bouthellier
Department of Mathematics
University of Tennessee
615 McCallie Avenue
Chattanooga, TN 37403

ABSTRACT

A simple technique will be derived for the simultaneous coefficient assignment and simultaneous stabilization of a r-tuple of m-input p-output linear time-varying systems where $r < m/p$. It will be noted that the systems we consider are not required to be slowly-varying. All our results are constructive.

1. INTRODUCTION.

The problems of stabilization and coefficient assignment of linear time-invariant plants by a linear time-invariant output feedback compensator have been among the most important problems in systems theory. However, since many of the systems considered in practice are time-varying, it is important to generalize results well-known in time-invariant system theory (see [1],[2],[3],[4]) to the time-varying case. Hence in section 2 we shall describe a scheme to simultaneously coefficient assign and simultaneously stabilize a r-tuple of m-input p-output linear time-varying systems by a stable nonswitching time-varying output feedback compensator when the inequality $r < m/p$ is satisfied. It will be noted that the simultaneous stabilization problem has been shown to be a prelude to the more general problem of stabilizing a structured multiparameter family of plants [1],[2],[3],[4].

In this paper, as we require $r < m/p$, we shall consider the case where the number of inputs m is greater than the number of outputs p. The case where $m \leq p$ has been examined in [5] using a recursive algorithm.

The simultaneous coefficient assignment and simultaneous stabilization problems are described in detail in section 2. We will then obtain sufficiency conditions under which the two problems can be solved for the class of time-varying systems that can be realized as finite dimensional Autoregressive Moving Average (ARMA) models. We remark that the methods presented in this paper allow us to explicitly construct the compensators and are free of the requirement that the plant parameters be slowly varying. Since obtaining the precise values of the plant parameters for a time-varying system is often difficult, the robustness of the closed-loop systems are ascertained by appealing to a notion of generalized stability regions described by the authors in [6] (see also [7]).

In section 3 the coefficient assignment and stabilization schemes described in section 2 will be illustrated by deriving an output feedback compensator which will deadbeat a 2-input 1-output time-varying system. In section 4 concluding remarks will be made.

2. A COMPENSATOR DESIGN FOR SIMULTANEOUS COEFFICIENT ASSIGNMENT AND SIMULTANEOUS STABILIZATION

In this section we will consider the problems of simultaneous coefficient assignment and simultaneous stabilization for a r-tuple of m-input p-output linear time-varying systems. For r < m/p it will be shown that, under a technical condition described in theorem 2.9, it is possible to simultaneously cofficient assign (and hence stabilize) the resulting r-tuple of closed-loop systems by a stable nonswitching time-varying compensator. The objective of the compensator will be to assign the left hand side of each of the r resulting closed-loop systems.

We first define the class of systems we shall consider.

Definition 2.1: Let $IR^{i\times j}$ be the space of $i \times j$ matrices whose elements belong to IR. A system of the form

$$y(k) + \sum_{i=1}^{l} D(i,k)y(k-i) = \sum_{i=1}^{l} N(i,k)\nu(k-i) \tag{2.1}$$

where $D(i,k) \in IR^{p\times p}$ and $N(i,k) \in IR^{p\times m}$ $i = 1,2,\ldots,l$ $\forall k \in Z$ is defined to be a m-input p-output ARMA model of lag l.

For the sake of notation we shall define $B_l(m,p)$ to be the space of all $p \times m$ ARMA models of lag l which satisfy i.) $\nu(i) = \underline{0} \in IR^m$ and $y(i) = \underline{0} \in IR^p$ $\forall i < 0$ and ii) $\| D(i,\cdot) \|$ and $\| N(i,\cdot) \|$ $i = 1,2,\ldots,l$ are uniformly bounded functions of time (where $\| \cdot \|$ denotes the standard Euclidean norm).

Note: By considering the parameter vector of an ARMA model $(D(1,k), D(2,k),\ldots, D(l,k); N(1,k),\ldots,N(l,k))$ as a column vector in $IR^{l(p^2+pm)}$ for each time k, time-varying systems of the form (2.1) can be viewed as elements of the product space $\prod_{j=-\infty}^{\infty} IR^{l(p^2+pm)}$. We shall give this space the product topology [8].

Whereas our results will be derived for strictly proper systems, analogous results will be stated for the proper case. The proof techniques being similiar, their derivations are omitted.

The simultaneous coefficient assignment problem is now described.

Problem 2.2 (Simultaneous Coefficient Assignment): Given a r-tuple of $p \times m$ systems

$$y(k) + \sum_{i=1}^{l} D_j(i,k)y(k-i) = \sum_{i=1}^{l} N_j(i,k)\nu(k-i) \tag{2.2}$$

$j = 1, 2, \ldots, r$ belonging to $B_l(m,p)$, and denoted by $\{G_j\}_{j=1}^r$, find a stable $m \times p$ compensator $C(k)$

$$u^*(k) + \sum_{i=1}^{q} D(i,k)u^*(k-i) = \sum_{i=0}^{q} N(i,k)y(k-i) \tag{2.3}$$

$$\nu(k) = u(k) - u^*(k)$$

of finite lag q which assigns the coefficients $\Delta_j(i,k)$ of the equations of the closed-loop systems (G_j, C) described as

$$y(k) + \sum_{i=1}^{l+q} \Delta_j(i,k)y(k-i) = \sum_{i=1}^{l+q} r_j(i,k)u(k-i) \tag{2.4}$$

$j = 1, 2, \ldots, r$.

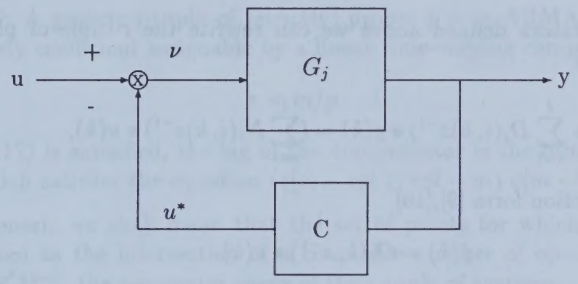


Figure 2.1: Closed-loop system corresponding to (G_j, C).

In this paper we shall assume that the coefficients $\Delta_j(i,k)$ in (2.4) are uniformly bounded functions of time. It will be noted that problem 2.2 is important in tracking, model matching, and stabilizability problems for plants which change operating conditions.

As a corollary of problem 2.2 we shall also solve

Problem 2.3 (Simultaneous Stabilization): Given a r-tuple of systems (2.2), find a stable compensator (2.3) such that for bounded input sequences $u(\cdot)$, all the signals in the closed-loop systems $(G_j, C)\ j = 1, 2, \ldots, r$ remain bounded.

To solve problems 2.2 and 2.3 we will use an operational algebra for time-varying systems which is described as follows: Where $a(k) \in IR^{p \times p}\ \forall k \in Z$ define the operator $z^{-l}\bullet$ [9],[10] by

$$z^{-l} \bullet a(k) = a(k-l) \tag{2.5}$$

and the operator $z^{-l}\circ$ [9],[10] by

$$z^{-l} \circ a(k) = a(k-l)z^{-l} \tag{2.6}$$

The following proposition, which lists the key properties of the above operators, may be verified by direct computation and hence the proof is omitted.

Proposition 2.4: Where $IR^{p\times m}[z^{-1}]$ is defined to be the set of power series in z^{-1} with coefficients taking values in $IR^{p\times m}$:
i) If $D(k,z^{-1}) \equiv I + \sum_{i=1}^{l} D(i,k)z^{-i} \in IR^{p\times p}[z^{-1}]$, there exists a unique inverse $D^{-1}(k,z^{-1}) \in IR^{p\times p}[z^{-1}]$ such that

$$D^{-1}(k,z^{-1}) \circ D(k,z^{-1}) = D(k,z^{-1}) \circ D^{-1}(k,z^{-1}) = I \ \forall k \in Z \tag{2.7}$$

ii) Where $A_i(k,z^{-1}) \in IR^{p_i\times p_{i+1}}[z^{-1}]\, i = 1,2,3$ and $y(k) \in IR^{p_4}$ where p_i $i = 1,2,3$, and 4 are finite positive integers

$$[A_1(k,z^{-1}) \circ ((A_2(k,z^{-1}) \circ A_3(k,z^{-1}))] \bullet y(k)$$
$$= A_1(k,z^{-1}) \bullet [A_2(k,z^{-1}) \bullet (A_3(k,z^{-1}) \bullet y(k))] \tag{2.8}$$

iii) 'o' is associative.

Using the operators defined above we can rewrite the r-tuple of plants $\{G_j\}_{j=1}^{r}$ defined by (2.2) as

$$(I + \sum_{i=1}^{l} D_j(i,k)z^{-i}) \bullet y(k) = (\sum_{i=1}^{l} N_j(i,k)z^{-i}) \bullet \nu(k), \tag{2.9}$$

or, in transfer function form [9],[10]

$$y(k) = G_j(k,z^{-1}) \bullet \nu(k) \tag{2.10}$$

where

$$G_j = D_j^{-1}(k,z^{-1}) \circ N_j(k,z^{-1}) \tag{2.11}$$

and

$$D_j(k,z^{-1}) = I + \sum_{i=1}^{l} D_j(i,k)z^{-i}; N_j(k,z^{-1}) = \sum_{i=1}^{l} N_j(i,k)z^{-i}. \tag{2.12}$$

Similiarly, the compensator C given by (2.3) may be written as

$$u^{\star}(k) = C(k,z^{-1}) \bullet y(k) \tag{2.13}$$

where, writing the compensator in right-matrix form

$$C = \bar{N}(k,z^{-1}) \circ \bar{D}^{-1}(k,z^{-1}) \tag{2.14}$$

and

$$\bar{N}(k,z^{-1}) = \sum_{i=0}^{q} \bar{N}(i,k)z^{-i}; \bar{D}(k,z^{-1}) = I + \sum_{i=1}^{q} \bar{D}(i,k)z^{-i} \tag{2.15}$$

Whenever convenient, in the future we shall write D, $\bar{D}$, N, and $\bar{N}$ for $D(k,z^{-1})$, $\bar{D}(k,z^{-1})$, $N(k,z^{-1})$, and $\bar{N}(k,z^{-1})$ respectively.

With the r-tuple of plants written as $G_j = D_j^{-1} \circ N_j$ and the compensator written as $C = \bar{N} \circ \bar{D}^{-1}$ it has been shown [10] that the equations of the closed-loop systems

(G_j, C) as defined by Figure 2.1 are given as follows:

Theorem 2.5: The equations of the closed-loop systems (G_j, C) are given by

$$y(k) = [\bar{D} \circ (D_j \circ \bar{D} + N_j \circ \bar{N})^{-1} \circ N_j] \bullet u(k) \tag{2.16}$$

$j = 1, 2 \ldots, r$.

Remark 2.6: Note how equation 2.16 generalizes the standard time-invariant result. The '$\circ$' operator is seen to replace multiplication.

As it has been shown [11] that the coefficients of the closed-loop systems (G_j, C) are nonlinear in the plant and compensator parameters, it appears that the coefficient assignment problem is difficult to solve. However, for the class of systems defined below, the problem may be completely resolved.

Theorem 2.7: A generic r-tuple of (strictly) proper $p \times m$ ARMA systems of lag l is simultaneously coefficient assignable by a linear time-varying compensator provided

$$r < m/p \tag{2.17}$$

Provided (2.17) is satisified, the lag of the compensator is the minimum non-negative integer q which satisfies the equation $(q[m - rp] \geq rpl - m)$ $q[m - rp] \geq rpl$.

Note: By generic we shall mean that the set of plants for which theorem 2.7 holds can be written as the intersection of a countable number of open and dense sets in $\prod_{j=-\infty}^{\infty} I\!R^{rl(p^2+pm)}$, the parameter space of the r-tuple of systems.

The proof of theorem 2.7 will give a constructive way of coefficient assigning time-varying systems. Theorem 2.9 will then solve the coefficient assignment problem by giving conditions under which coefficient assignment is possible with a stable compensator. This will then allow us to solve the simultaneous stabilization problem in corollary 2.10.

Proof (of theorem 2.7): We shall prove this result for the strictly proper case. The proof of the proper case, being similiar, is omitted.

By theorem 2.5, the equations of the closed-loop systems (G_j, C) where $G_j = D_j^{-1} \circ N_j$ and $C = \bar{N} \circ \bar{D}^{-1}$ are given by

$$y = [\bar{D} \circ (D_j \circ \bar{D} + N_j \circ \bar{N})^{-1} \circ N_j] \bullet u \tag{2.18}$$

$j = 1, 2, \ldots, r$. Defining $\bar{D} \equiv I$ the equations of the closed-loop systems are given by

$$y = [(D_j + N_j \circ \bar{N})^{-1} \circ N_j] \bullet u, \tag{2.19}$$

which may be written as

$$(D_j + N_j \circ \bar{N}) \bullet y = N_j \bullet u. \tag{2.20}$$

To simultaneously coefficient assign (2.20) we want, via the coefficients of $u^*(k) = \bar{N} \bullet y = \sum_{i=0}^{q} \bar{N}(i,k) y(k-i)$, to assign the coefficients $\Delta_j(1,k), \Delta_j(2,k), \ldots, \Delta_j(l+q,k)$ of

$$D_j + N_j \circ \bar{N} = I + \sum_{i=1}^{l+q} \Delta_j(i,k) z^{-i}. \tag{2.21}$$

Subtracting D_j from both sides, (2.21) implies

$$(\sum_{i=1}^{l} N_j(i,k)z^{-i}) \circ (\sum_{j=0}^{q} \bar{N}(j,k)z^{-j}) = \sum_{i=1}^{l+q} \Delta_j(i,k)z^{-i} - \sum_{i=1}^{l} D_j(i,k)z^{-i} \tag{2.22}$$

We shall coefficient assign the r-tuple of systems by finding $\bar{N}(0,k)$, $\bar{N}(1,k),\ldots,$ $\bar{N}(q,k)$ which satisfy (2.22) for $j = 1,2,\ldots,r$.

Expanding the l.h.s. of (2.22), equating like powers of z^{-1}, and combining the r equalities yield the system of equations

$$M(k)v(k) = \Delta(k) \tag{2.23}$$

where

$$M(k) = \begin{bmatrix} N^r(1,k+1) & O^r & \ldots & O^r \\ N^r(2,k+2) & N^r(1,k+2) & & O^r \\ \vdots & N^r(2,k+3) & & \vdots \\ \vdots & \vdots & & \vdots \\ N^r(l,k+l) & \vdots & & O^r \\ O^r & N^r(l,k+l+1) & & N^r(1,k+q+1) \\ O^r & O^r & & N^r(2,k+q+2) \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ O^r & O^r & \ldots & N^r(l,k+q+l) \end{bmatrix} \tag{2.24}$$

$$N^r(i,j) \equiv \text{col}\,[N_1(i,j), N_2(i,j),\ldots,N_{r-1}(i,j), N_r(i,j)] \tag{2.25}$$

$O^r$is the zero matrix in $IR^{pr\times m}$,

$$v(k) \equiv \text{col}[\bar{N}(0,k), \bar{N}(1,k+1),\ldots,\bar{N}(q,k+q)] \tag{2.26}$$

and

$$\Delta(k) \equiv \text{col}\,[[\Delta(1,k+1) - D(1,k+1)]^r, [\Delta(2,k+2) - D(2,k+2)]^r,\ldots,$$

$$[\Delta(l,k+l) - D(l,k+l)]^r, \Delta^r(l+1,k+l+1),\ldots,\Delta^r(l+q,k+l+q)] \tag{2.27}$$

The $[\,]^r$ and $\Delta^r(\cdot)$ terms in (2.27) are defined similiarly to those of (2.25). From (2.20)–(2.23) we see that we can coefficient assign the l.h.s. of the closed-loop systems (2.20)–(2.21) provided (2.23) is solvable.

It is easy to show that $M(k)$ is a $rp(l+q)\times(q+1)m$ matrix. It has been shown [11] that $M(k)$ is generically of full rank where q is chosen to be the smallest non-negative integer satisfying

$$q[m-rp] \geq rpl - m. \tag{2.28}$$

Hence, for a generic r-tuple of $p \times m$ plants (2.23) is solvable, and thus coefficient assignment is possible, provided $r < m/p$.

(Q.E.D.)

Theorem 2.7 shows that a generic r-tuple of systems

$$y(k) + \sum_{i=1}^{l} D_j(i,k)y(k-i) = \sum_{i=1}^{l} N_j(i,k)\nu(k-i) \tag{2.29}$$

$j = 1, \ldots, r$ is simultaneously coefficient assignable by a compensator of the form

$$u^{\star}(k) = \sum_{i=0}^{q} \bar{N}(i,k)y(k-i) \tag{2.30}$$

where i.) q is the smallest non-negative integer which satisfies

$$q[m-rp] \geq rpl - m \tag{2.31}$$

and ii) the vector of coefficients $v(k) \equiv \text{col}[\bar{N}(0,k), \bar{N}(1,k+1), \ldots, \bar{N}(q,k+q)]$ satisfies (2.23).

In order to solve the coefficient assignment problem (and the simultaneous stabilization problem) we require the compensator (2.30) to be stable. Hence we require the coefficients $\bar{N}(0,k) \ldots, \bar{N}(q,k)$ to be uniformly bounded functions of time. Thus we require the following well-known proposition.

Proposition 2.8[12]: Let z be a m vector and A be a $m \times n$ matrix of rank m. The n vector x which satisifies $Ax = z$ and mimimizes $x^T x$ is given by

$$x = A^T(AA^T)^{-1}z \tag{2.32}$$

We now give the main result of this paper.

Theorem 2.9: A r-tuple of $p \times m$ systems $\{G_j\}_{j=1}^{r} \in B_l(m,p)$ which satisfy i.) $r < m/p$ and ii.) $\det[M(k)M^T(k)] > \varepsilon$ for some $\varepsilon > 0$ independent of k is simultaneous coefficient assignable by a stable compensator (2.30) where q satisfies (2.31) and

$$\text{col}\,[\bar{N}(0,k), \bar{N}(1,k+1), \ldots, \bar{N}(q,k+q)] = M^T(k)[M(k)M^T(k)]^{-1}\Delta(k), \tag{2.33}$$

where $M(k)$ and $\Delta(k)$ are given by (2.24) and (2.27) respectively.

Proof: Follows directly from the above results.

As a corollary of theorem 2.9 we now solve the simultaneous stabilization problem.

Corollary 2.10: A r-tuple of $p \times m$ systems $\{G_j\} \in B_l(m,p)$ which satisfy conditions i.) and ii.) of theorem 2.9 are simultaneously stabilizable by a stable compensator.

Remark 2.11: Note that nowheres in our coefficient assignment or stabilization schemes did we require the coefficients of the systems to be slowly varying.

Remark 2.12: It is easy to show that if the compensator C, as constructed by theorem 2.9, makes the closed-loop systems exponentially stable, any initial conditions in the system will decay exponentially to 0.

Remark 2.13: Often systems are stabilized by assigning the coefficients $\Delta_j(i,k)$ so that

the l.h.s.'s of (2.20) are stable time-invariant operators. It will be noted that, via (2.23), that if we not have perfect information about the plant parameters, the l.h.s.'s will in fact be time-varying operators evolving in neighborhoods of stable time-invariant plants. However, it has been shown [6],[7] that a time-varying system evolving in a small neighborhood of a stable time-invariant plant is also stable. Hence our stabilization result is robust in the sense that small perturbations in the plants parameters will not affect the stability of the closed-loop systems.

3. EXAMPLE OF COEFFICIENT ASSIGNMENT

We shall consider designing a deadbeat control law for the time-varying system G given by

$$y(k) + d_1(k)y(k-1) + d_2(k)y(k-2) = [n_1^1(k)\ n_1^2(k)] \begin{bmatrix} \nu_1(k-1) \\ \nu_2(k-1) \end{bmatrix} \tag{3.1}$$

$$+[n_2^1(k)\ n_2^2(k)] \begin{bmatrix} \nu_1(k-2) \\ \nu_2(k-2) \end{bmatrix}$$

where

$$\begin{bmatrix} d_1(k) & d_2(k) \end{bmatrix} = \begin{cases} \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} & \text{k even} \\ & \text{k odd} \end{cases} \tag{3.2i}$$

and

$$\begin{bmatrix} n_1^1(k) & n_1^2(k) \end{bmatrix} = \begin{cases} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} & \text{k even} \\ & \text{k odd} \end{cases} \tag{3.2ii}$$

$$\begin{bmatrix} n_2^1(k) & n_2^2(k) \end{bmatrix} = \begin{cases} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} & \text{k even} \\ & \text{k odd} \end{cases} \tag{3.2iii}$$

By theorem 2.7, as p=1, m=2, l=2, and the number of plants r=1 (hence the coefficient assignment condition $r < m/p$ is satisfied), the 1×2 compensator C which will deadbeat (3.1) is given by

$$u^\star(k) = \sum_{i=0}^{q} \begin{bmatrix} \bar{n}_1(k) \\ \bar{n}_2(k) \end{bmatrix} y(k-i), \tag{3.3}$$

where q is the smallest nonnegative integer which satisfies $q[2 - 1(1)] \geq (1)(1)(2) - 2$, (so $q = 0$). Hence by theorem 2.9 the compensator is given by

$$u^\star(k) = \begin{bmatrix} \bar{n}_1(k) \\ \bar{n}_2(k) \end{bmatrix} y(k) \tag{3.4}$$

where

$$\begin{bmatrix} \bar{n}_1(k) \\ \bar{n}_2(k) \end{bmatrix} = M^T(k)[M(k)M^T(k)]^{-1}\Delta(k). \tag{3.5}$$

As

$$M(k) = \begin{bmatrix} n_1^1(k+1) & n_1^2(k+1) \\ n_2^1(k+2) & n_2^2(k+2) \end{bmatrix} \tag{3.6i}$$

and

$$\Delta(k) = \text{col}\begin{bmatrix} -d_1(k+1) & -d_2(k+2) \end{bmatrix} \tag{3.6ii}$$

it is easy to show

$$\begin{bmatrix} \bar{n}_1(k) \\ \bar{n}_2(k) \end{bmatrix} = M^{-1}(k)\Delta(k)$$

$$= \begin{cases} \begin{bmatrix} -4 \\ -2 \end{bmatrix} & \text{k even} \\ \begin{bmatrix} 0 \\ -1 \end{bmatrix} & \text{k odd} \end{cases} \tag{3.7}$$

Thus

$$u^*(k) = \begin{bmatrix} \bar{n}_1(k) \\ \bar{n}_2(k) \end{bmatrix} y(k) \tag{3.8}$$

where col $[\bar{n}_1(k)\,\bar{n}_2(k)]$ is given by (3.7) is a deadbeat control law for (3.1).

Using theorem 2.5 it is seen that the equation of the closed-loop system (G, C) is given by

$$y(k) = \begin{bmatrix} n_1^1(k) & n_1^2(k) \end{bmatrix} \begin{bmatrix} u_1(k-1) \\ u_2(k-1) \end{bmatrix} + \begin{bmatrix} n_2^1(k) & n_2^2(k) \end{bmatrix} \begin{bmatrix} u_1(k-2) \\ u_2(k-2) \end{bmatrix} \tag{3.9}$$

4. CONCLUSION

In this paper we have given conditions under which a r-tuple of $p \times m$ systems may be simultaneously stabilized and coefficient assigned by a stable nonswitching output feedback compensator which is easy to construct. It was noted that our technique is free of the requirement that the time-varying systems be slowly varying.

REFERENCES

[1] B.K. Ghosh, "An Approach to Simultaneous System Design, I. Semialgebraic Geometric Methods," SIAM J. Control and Optimization, Vol. 24, No. 3, May 1986, pp. 480–496.

[2] B.K. Ghosh, "An Approach to Simultaneous System Design, II. Nonswitching Gain and Dynamic Feedback Compensation by Algebraic Geometric Methods," SIAM J. Control and Optimization, Vol. 26, No. 4, July 1988, pp. 919–963.

[3] B.K. Ghosh,"Transcendental and Interpolation Methods in Simultaneous Stabilization and Simultaneous Partial Pole Placement Problems," SIAM J. Control and Optimization, Vol. 24, No. 4, July 1986, pp. 1091–1109.

[4] B.K. Ghosh and C.I. Byrnes, "Simultaneous Stabilization and Simultaneous Pole-Placement by Non-Switching Dynamic Compensation," IEEE Trans. on Automatic Control, Vol. AC-28, No. 6, June 1983, pp. 735–741.

[5] B. K. Ghosh and P. R. Bouthellier,"Simultaneous Coefficient Assignment of Discrete Time Multi-Input Multi-Output Linear Time-Varying Systems: A New Approach for Compensator Design," SIAM J. Control and Optimization, Vol. 31, No. 6, Nov. 1993.

[6] P.R. Bouthellier and B.K. Ghosh,"A Stability Theory of Linear Time-Varying Systems," Proc. 25th Annual Allerton Conference on Communication, Control, and Computing, 1987.

[7] C.A. Desoer,"Slowly Varying Discrete System $x_{i+1} = A_i x_i$," Electronic Letters, Vol. 6, No. 11, 1970, pp. 339-340.

[8] I.M. Singer and J.A. Thorpe, *Lecture Notes on Elementary Topology and Geometry*, New York: Scott, Foreman, Glenview, IL, 1967.

[9] E.W. Kamen, P.P. Khargonekar, and K.R. Poolla," A Transfer Function Approach to Linear Time-Varying Discrete-Time Systems," SIAM J. Control and Optimization Vol. 23, No. 4, July 1985, pp. 550–565.

[10] P.R. Bouthellier and B.K. Ghosh,"Robust Stabilization of Discrete-Time Single-Input Single-Output Linear Time-Varying Systems," Proc. of 27th IEEE Conf. on Decision and Control, 1988, Austin, Texas, pp. 39-44.

[11] P.R. Bouthellier, *Analysis and Design of Discrete-Time Linear Time-Varying Systems,* D.Sc . Dissertation, Washington University, St. Louis, MO, 1990.

[12] R.W. Brockett, *Finite Dimensional Linear Systems,* John Wiley and Sons, 1969.

Analysis and Design of Integral Controllers for Discrete Systems

John G. Kuschewski
Center for Neural Engineering
Tennessee State University
Nashville, TN 37209-1561

Stefen Hui
Dept. of Mathematical Sciences
San Diego State University
San Diego, CA 92812

Stanislaw H. Żak
School of Electrical Engineering
Purdue University
West Lafayette, IN 47907-1285

Abstract

We analyze an integral controller for linear, time-invariant, single-input single-output discrete dynamic systems. We first describe an integral controller and then give a necessary and sufficient condition under which the state of the controller and the state of the dynamic system converge, that is, steady-state tracking is achieved. Conditions under which the controller state and the dynamic system state diverge are also investigated. In particular, we analyze how the choice of the integral controller gain influences convergence, and we compute the gain interval securing stable controller operation. We then apply this controller to dc servomechanism position control. Computer simulation as well as experimental laboratory results are given for this application.

1 Introduction

We consider a linear, time-invariant, single-input single-output discrete dynamic system (plant) modeled by

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{b}u_k \\ y_k &= \boldsymbol{c}\boldsymbol{x}_k + du_k \end{aligned}$$

and an integral controller described by

$$w_{k+1} = w_k + Ke_k \tag{1}$$
$$u_k = w_k \tag{2}$$

where k = the time index, $w_k \in \mathbb{R}$ is the controller state at time k, $\boldsymbol{x}_k \in \mathbb{R}^n$ is the plant state vector at time k, $u_k = w_k \in \mathbb{R}$ is the plant input at time k, $y_k \in \mathbb{R}$ is the plant output at time k, $y_d \in \mathbb{R}$ is the desired plant output, $e_k = y_d - y_k$ is the error at time k, $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, $\boldsymbol{b}, \boldsymbol{c}^T \in \mathbb{R}^n$, $d \in \mathbb{R}$, and $K \in \mathbb{R}$ is the integral controller gain.

The matrix $\boldsymbol{A}$ is assumed to be asymptotically stable, that is, the eigenvalues of $\boldsymbol{A}$ are contained in the open unit disk. Note that the integral controller described by (1) and (2) is similar to the one-sample approximation of an analog integrator [1, p. 119].

The closed-loop control system is depicted in Fig. 1.

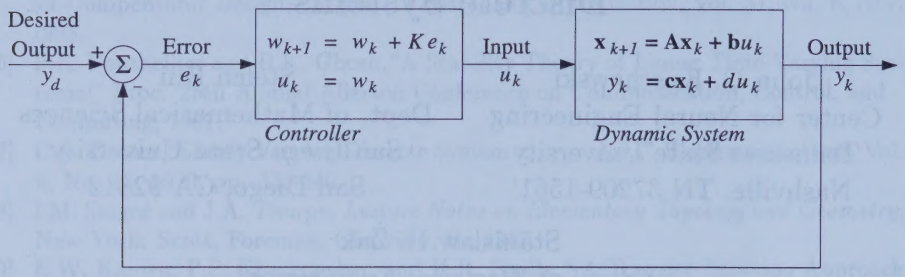


Figure 1: A closed-loop system with an integral controller for a linear, time-invariant, single-input single-output discrete dynamic system.

2 Analysis of the Closed-Loop Control System

Using the definitions of u_k and e_k we obtain

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{b}w_k, \tag{3}$$

and

$$\begin{aligned} w_{k+1} &= w_k + K(y_d - y_k) \\ &= w_k + K(y_d - \boldsymbol{c}\boldsymbol{x}_k - dw_k) \\ &= (-K\boldsymbol{c})\boldsymbol{x}_k + (1 - Kd)w_k + Ky_d. \end{aligned} \tag{4}$$

In matrix form, (3) and (4) become

$$\begin{bmatrix} \boldsymbol{x}_{k+1} \\ w_{k+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{b} \\ -K\boldsymbol{c} & 1 - Kd \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_k \\ w_k \end{bmatrix} + Ky_d \begin{bmatrix} \boldsymbol{0} \\ 1 \end{bmatrix}. \tag{5}$$

The goal is to find a range for the integral gain $K \neq 0$ so that given any y_d the controller state and the dynamic system state converge, i.e.,

$$\lim_{k\to\infty} w_k = w_\infty \qquad \text{and} \qquad \lim_{k\to\infty} \boldsymbol{x}_k = \boldsymbol{x}_\infty. \tag{6}$$

If the limits in (6) exist, it then follows from

$$w_{k+1} - w_k = K(y_d - y_k)$$

that

$$\lim_{k\to\infty} y_k = y_d$$

since $K \neq 0$.

Suppose now that the limits in (6) exist. Then,

$$\boldsymbol{x}_\infty = \boldsymbol{A}\boldsymbol{x}_\infty + \boldsymbol{b}w_\infty,$$

and

$$y_d = \boldsymbol{c}\boldsymbol{x}_\infty + dw_\infty.$$

Since we have assumed that $\boldsymbol{A}$ is asymptotically stable, $(\boldsymbol{I}_m - \boldsymbol{A})^{-1}$ exists and thus

$$y_d = [\boldsymbol{c}(\boldsymbol{I}_m - \boldsymbol{A})^{-1}\boldsymbol{b} + d]w_\infty.$$

Since y_d is arbitrary, a necessary condition for the limits in (6) to exist is

$$\boldsymbol{c}(\boldsymbol{I}_m - \boldsymbol{A})^{-1}\boldsymbol{b} + d \neq 0. \tag{7}$$

It turns out that (7) is also a sufficient condition for the limits in (6) to exist.

In further discussion, we use the following notation. Let $\sigma(\boldsymbol{M})$ denote the set of eigenvalues of the square matrix $\boldsymbol{M}$ and let Δ be the open unit disk. We then have the following theorem [2]:

Theorem 1 *If $\sigma(\boldsymbol{A}) \subset \Delta$, then a necessary and sufficient condition for the existence of $K \neq 0$ so that given any y_d the limits in (6) exist is that $\boldsymbol{c}(\boldsymbol{I}_m - \boldsymbol{A})^{-1}\boldsymbol{b} + d \neq 0$.*

Proof

See Hui and Żak [2] or Kuschewski et al. [3].

□

The following three lemmas [2] are used in the proof of Theorem 1, and are restated here. However, before we can restate these lemmas, we need to define the following notation. Let

$$\boldsymbol{G}_K = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{b} \\ -K\boldsymbol{c} & 1 - Kd \end{bmatrix},$$

and

$$Q(\lambda) = \boldsymbol{c}(\lambda \boldsymbol{I}_m - \boldsymbol{A})^{-1}\boldsymbol{b} + d.$$

Let $\mathrm{adj}(\boldsymbol{M})$ denote the standard adjoint of the square matrix $\boldsymbol{M}$ (see, e.g., [4, p. 25]). Recall that if $\boldsymbol{M}$ is nonsingular, then $\mathrm{adj}(\boldsymbol{M}) = \det(\boldsymbol{M})\boldsymbol{M}^{-1}$.

Lemma 1 *1. If $\sigma(\boldsymbol{M}) \subset \Delta$, then $\lim_{k\to\infty} \boldsymbol{M}^k = \boldsymbol{0}$.*

2. If $\sigma(\boldsymbol{M}) \subset \Delta \cup \{1\}$, then $\lim_{k\to\infty} \boldsymbol{M}^k$ exists if and only if the dimension of the eigenspace corresponding to the eigenvalue 1 is equal to the multiplicity of the eigenvalue 1, i.e., the geometric multiplicity of the eigenvalue 1 is equal to its algebraic multiplicity.

Proof

See Hui and Żak [2] or Kuschewski et al. [3].

□

Lemma 2 *The characteristic polynomial of* $\boldsymbol{G}_K$ *is*

$$\begin{aligned}\det[\lambda \boldsymbol{I}_{n+1} - \boldsymbol{G}_K] &= P_K(\lambda) \\ &= (\lambda - 1 + Kd)\det(\lambda \boldsymbol{I}_n - \boldsymbol{A}) + K\boldsymbol{c}[\mathrm{adj}(\lambda \boldsymbol{I}_n - \boldsymbol{A})]\boldsymbol{b}.\end{aligned}$$

Proof

See Hui and Żak [2] or Kuschewski et al. [3].

□

Lemma 3 *If* $\sigma(\boldsymbol{A}) \subset \Delta$, *then there is an interval* I *of the form* $(0, K_0)$ *or of the form* $(-K_0, 0)$, *depending on whether* $Q(1) = \boldsymbol{c}(\boldsymbol{I}_n - \boldsymbol{A})^{-1}\boldsymbol{b} + d$ *is positive or negative, respectively, so that for* $K \in I$,

$$\lim_{k\to\infty} \boldsymbol{G}_K^k \tag{8}$$

and

$$\lim_{k\to\infty} \sum_{i=0}^{k-1} \boldsymbol{G}_K^i \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{9}$$

both exist.

Proof

See Hui and Żak [2] or Kuschewski et al. [3].

□

3 Case Study: DC Servomechanism Control

We now implement the proposed integral controller on a permanent magnet armature-controlled dc servomechanism, manufactured by Electro-Craft Corporation. We first develop parameterized state space and transfer function models of the servomechanism. We next combine experimental servomechanism measurements with these models to generate experimental servomechanism models. Then, we model the overall control system, compute the interval I for correct controller operation, and present the simulation and experimental results.

3.1 System Modeling

A schematic diagram of the permanent magnet armature-controlled dc servomechanism is given in Fig. 2. There is one input, the input voltage e_{in}, and there are two outputs, the velocity output voltage e_v and the position output voltage e_p.

If we choose the state variables

$$x_1 = \theta_m,\ x_2 = \frac{d\theta_m}{dt} = \omega_m,\ x_3 = I_a,$$

the system model can be represented in state space format as:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\dfrac{B_e}{J_e} & \dfrac{K_m}{J_e} \\ 0 & -\dfrac{K_b}{L_a} & -\dfrac{R_a}{L_a} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dfrac{K_{pa}K_a}{L_a} \end{bmatrix} e_{in},$$

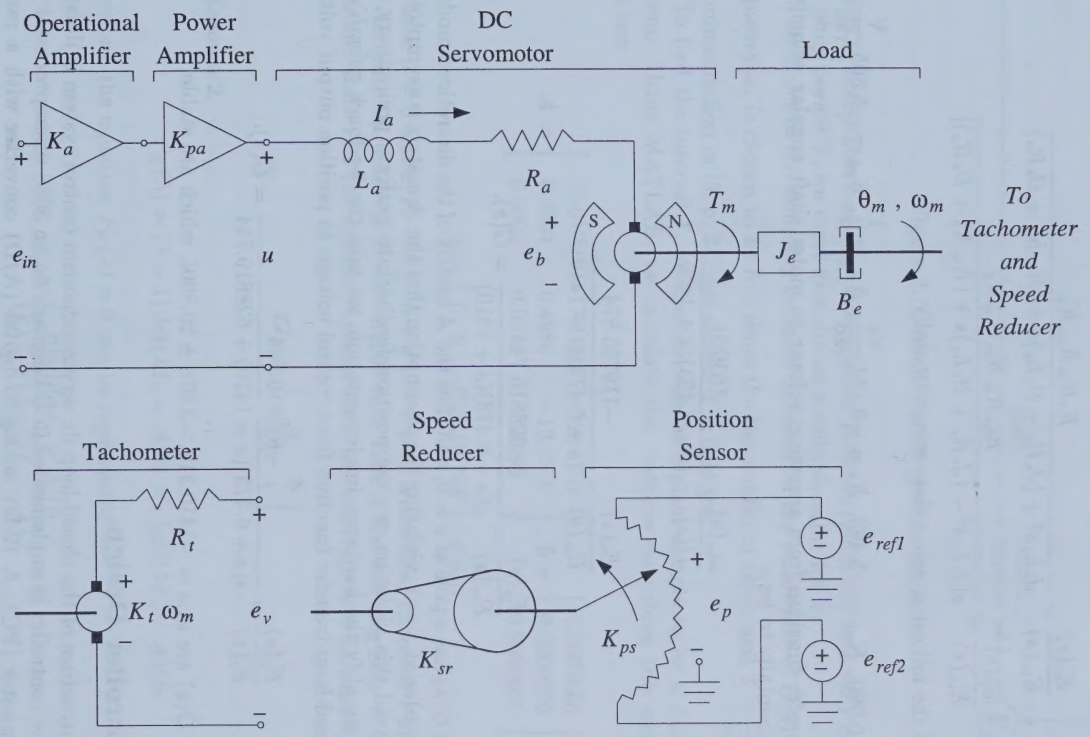


Figure 2: Schematic diagram of the permanent magnet armature-controlled dc servomechanism.

$$e_v = \begin{bmatrix} 0 & K_t & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \; e_p = \begin{bmatrix} K_{ps}K_{sr} & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}.$$

For a detailed derivation of this state space model, we refer the reader to Kuschewski, et al. [3]. From the system model in state space form, we obtain the following transfer functions:

$$\frac{E_v(s)}{E_{in}(s)} = \frac{K_t K_m K_{pa} K_a}{J_e L_a s^2 + (J_e R_a + B_e L_a)s + (K_m K_b + B_e R_a)},$$

$$\frac{E_p(s)}{E_{in}(s)} = \frac{K_{ps} K_{sr} K_m K_{pa} K_a}{s[J_e L_a s^2 + (J_e R_a + B_e L_a)s + (K_m K_b + B_e R_a)]}.$$

We found the following gain values experimentally:

$$K_a = -0.990, \; K_{pa} = -5.469, \; K_t = -0.144 \, \frac{\text{Vs}}{\text{rad}}, \; K_{sr} = \frac{1}{9}, \; K_{ps} = -6.805 \, \frac{\text{V}}{\text{rad}}.$$

Also, the transfer function from armature voltage to motor shaft angular velocity was found experimentally to be

$$\frac{\omega_m(s)}{U(s)} = \frac{153600}{(s + 6.473)(s + 1410)}.$$

Thus,

$$\frac{E_v(s)}{E_{in}(s)} = \frac{-119755.874}{(s + 6.473)(s + 1410)},$$

and

$$\frac{E_p(s)}{E_{in}(s)} = \frac{-628810.744}{s(s + 6.473)(s + 1410)} = G(s).$$

Note that $G(s)$ has a pole at $s = 0$, and thus the $\boldsymbol{A}$ matrix of the discretized model will have an eigenvalue at $\lambda = 1$, violating the assumption that the $\boldsymbol{A}$ matrix is asymptotically stable. To avoid this violation, we use position feedback to perform preliminary stabilization. To simplify the hardware implementation, we take the feedback gain $K_f = 1$. Thus the closed-loop transfer function from control voltage to position output voltage is

$$\frac{E_p(s)}{E_c(s)} = \frac{-628810.744}{s(s + 6.473)(s + 1410) + 628810.744} = \widehat{G}(s).$$

The poles of $\widehat{G}(s)$ are at $s = -1410.318, -3.078 \pm 20.890i$, which are stable.

3.2 Controller Design

The overall structure of the closed-loop dc servomechanism control system is illustrated in Fig. 3. The controller is implemented in C language on an 8088 microprocessor based personal computer (PC). A 12-bit analog-to-digital (A/D) converter with a range of $\pm$50V (Sample) is used to input the position output voltage from the position sensor to the controller, and a 12-bit digital-to-analog (D/A) converter with a range of $\pm$10V (Hold) is used to output the control voltage from the controller to the operational amplifier.

The goal of the integral controller design is to have the motor shaft angular position θ_m track a single step input or a series of step inputs while minimizing both oscillation and settling time. We must first find the interval I, and then choose a $K \in I$ that

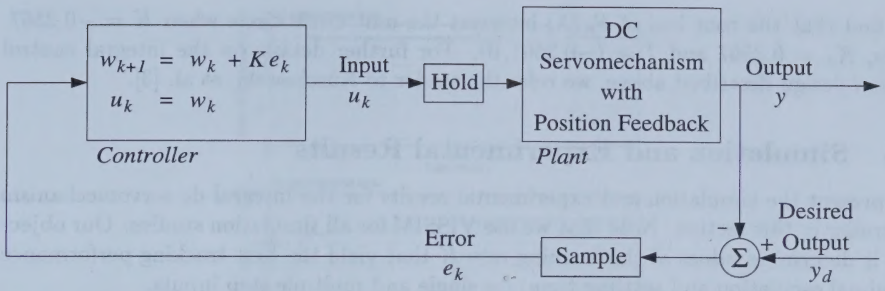


Figure 3: Overall control system structure.

results in acceptable tracking performance. However, since the interval I depends on the sampling period T, we must first choose a suitable value for T.

The choice of the sampling period T is based on the bandwidth of $\widehat{G}(s)$: the sampling frequency ω_s is chosen to be five times the bandwidth of $\widehat{G}(s)$, and $\mathrm{T} = \dfrac{2\pi}{\omega_s}$, following a recommendation in [5, p. 38]; we choose T = 0.04 sec.

To find the interval I, we consider the hold-plant-sample to be a discrete dynamic system. Using MATLAB, we calculate the matrices describing this system for T = 0.04 sec:

$$\boldsymbol{A} = \begin{bmatrix} -0.0120 & -17.1265 & -226.7016 \\ 0.0004 & 0.4986 & -13.8360 \\ 0.0000 & 0.0315 & 0.6995 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 0.0003605 \\ 0.0000220 \\ 0.0000005 \end{bmatrix},$$

$$\boldsymbol{c} = \begin{bmatrix} 0 & 0 & -628810.744 \end{bmatrix}, \quad d = 0.$$

We have $n = 3$ and $\boldsymbol{c}(\boldsymbol{I}_n - \boldsymbol{A})^{-1}\boldsymbol{b} + d = -1 \neq 0$. Therefore, Theorem 1 applies, and we can compute the interval $I = (-K_0, 0)$ such that if the learning rate $K \in I$, then the controller state and the dynamic system state will converge (see Lemma 3). We have

$$\boldsymbol{G}_K = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{b} \\ -K\boldsymbol{c} & 1 \end{bmatrix}.$$

By Lemma 2,

$$P_K(\lambda) = (\lambda - 1)\det(\lambda \boldsymbol{I}_n - \boldsymbol{A}) + K\boldsymbol{c}\,[\mathrm{adj}(\lambda \boldsymbol{I}_n - \boldsymbol{A})]\,\boldsymbol{b}.$$

Therefore, the equation $P_K(\lambda) = 0$ can be represented as

$$1 + K\left(\frac{\boldsymbol{c}\,[\mathrm{adj}(\lambda \boldsymbol{I}_n - \boldsymbol{A})]\,\boldsymbol{b}}{(\lambda - 1)\det(\lambda \boldsymbol{I}_n - \boldsymbol{A})}\right) = 0,$$

which evaluates to

$$1 + K\left(\frac{-0.3005\lambda^2 - 0.2950\lambda - 0.0002}{(\lambda - 1)\lambda(\lambda^2 - 1.1861\lambda + 0.7818)}\right) = 0.$$

We find that the root loci of $P_K(\lambda)$ intersect the unit circle circle when $K = -0.2567$. Thus, $K_0 = 0.2567$ and $I = (-0.2567, 0)$. For further details on the integral control system design described above, we refer the reader to Kuschewski, et al. [3].

3.3 Simulation and Experimental Results

We present the simulation and experimental results for the integral dc servomechanism controller in this section. Note that we use VISSIM for all simulation studies. Our objective is determine values of the learning rate K that yield the best tracking performance (minimal oscillation and settling time) for single and multiple step inputs.

We begin with a single step input. The desired $\theta_m(t) = 135h(t - 0.1)$ degrees; thus $e_{in}(t) = -1.7813h(t-0.1)$ Volts ($h(t)$ is the unit step function). From multiple simulation trials, we determined that $K = -0.15$ gave the best simulation tracking performance. The simulation results for $K = -0.15$ are given in Fig. 4, while the experimental results for $K = -0.15$ are given in Fig. 5. Note that the oscillations in the experimental data

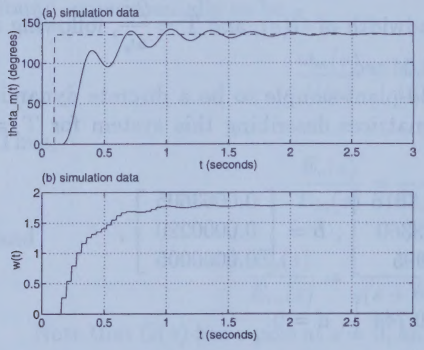


Fig. 4

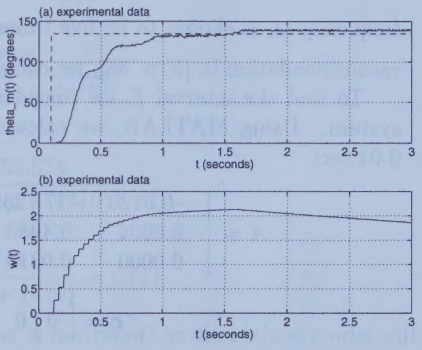


Fig. 5

Figure 4: Plot of (a) $\theta_m(t)$ versus t and (b) $w(t)$ versus t, for $e_{in}(t) = -1.7813h(t - 0.1)$ Volts; simulation data for $K = -0.15$. The dashed line in (a) represents the desired $\theta_m(t)$.

Figure 5: Plot of (a) $\theta_m(t)$ versus t and (b) $w(t)$ versus t, for $e_{in}(t) = -1.7813h(t - 0.1)$ Volts; experimental data for $K = -0.15$. The dashed line in (a) represents the desired $\theta_m(t)$.

are damped. This damping effect, due to unmodeled friction in the servomechanism that occurs at low motor shaft angular velocity, causes the experimental interval I_e to be larger than the theoretical interval I. From multiple experimental trials, we found $I_e = (-0.5428, 0)$. We also found that $K = -0.26$ gave the best experimental tracking performance; see Fig. 6 for the experimental results.

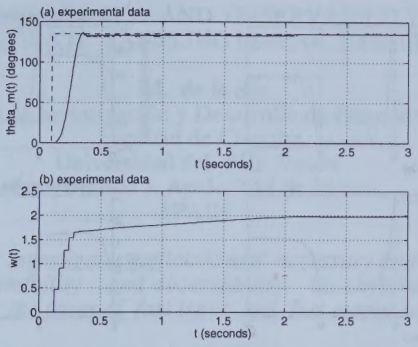


Figure 6: Plot of (a) $\theta_m(t)$ versus t and (b) $w(t)$ versus t, for $e_{in}(t) = -1.7813h(t-0.1)$ Volts; experimental data for $K = -0.26$. The dashed line in (a) represents the desired $\theta_m(t)$.

We end with a staircase input. The desired $\theta_m(t)$ is

$$\theta_m(t) = \begin{cases} 0°, & 0 \leq t < 0.1 \\ 90°, & 0.1 \leq t < 2 \\ -45°, & 2 \leq t < 4 \\ 135°, & 4 \leq t < 6 \\ -90°, & 6 \leq t < 8 \\ 45°, & 8 \leq t \leq 10 \end{cases}.$$

Thus

$$\begin{aligned} e_{in}(t) &= -1.1875h(t-0.1) + 1.7813h(t-2) - 2.3751h(t-4) \\ &\quad + 2.9688h(t-6) - 1.7813h(t-8) \text{ Volts.} \end{aligned} \tag{10}$$

The simulation results for $\theta_m(t)$ versus t and $w_1(t)$ versus t, for $e_{in}(t)$ given by (10) and $K = -0.15$, are presented in Fig. 7. From multiple experimental trials, we found that $K = -0.245$ gave the best experimental tracking performance. The experimental results for $\theta_m(t)$ versus t and $w_1(t)$ versus t, for $e_{in}(t)$ given by (10) and $K = -0.245$, are presented in Fig. 8.

4 Acknowledgements

The work of John G. Kuschewski, while at Purdue University, was supported in part by a David Ross Grant 690-1285-1557 from Purdue University.

References

[1] HOSTETTER, G. H., Digital Control System Design. Holt, Rinehart, and Winston, New York, N.Y., 1988

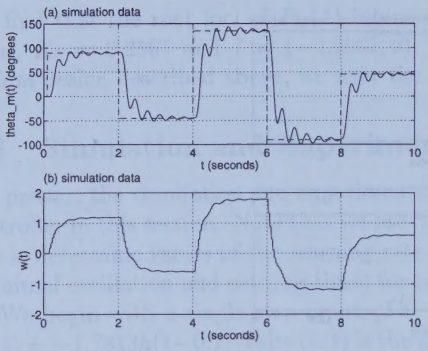


Fig. 7

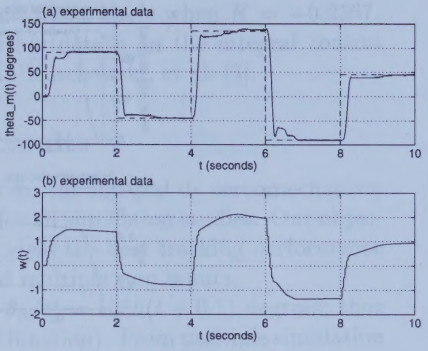


Fig. 8

Figure 7: Plot of (a) $\theta_m(t)$ versus t and (b) $w(t)$ versus t, for $e_{in}(t)$ given by (10); simulation data for $K = -0.15$. The dashed line in (a) represents the desired $\theta_m(t)$.

Figure 8: Plot of (a) $\theta_m(t)$ versus t and (b) $w(t)$ versus t, for $e_{in}(t)$ given by (10); experimental data for $K = -0.245$. The dashed line in (a) represents the desired $\theta_m(t)$.

[2] Hui, S. and Żak, S. H., 'Analysis of single perceptrons learning capabilities', in *Proceedings of the American Control Conference*, volume 1, pp. 809–814, Boston, MA., 1991

[3] Kuschewski, J. G., Hui, S., and Żak, S. H., 'On integral control of discrete dynamic systems', Submitted for publication

[4] Barnett, S., Introduction to Mathematical Control Theory. Clarendon Press, Oxford, England, 1975

[5] Ogata, K., Discrete-Time Control Systems. Prentice Hall, Englewood Cliffs, N.J., 1987

ON THE REACHABILITY AND OBSERVABILITY OF HYBRID MU LTIRATE SAMPLING LINEAR SYSTEMS

M. de la Sen
Instituto de Investigación y Desarrollo de Procesos IIDP
Facultad de Ciencias
Universidad del País Vasco
Leioa (Bizkaia). Aptdo. 644 de Bilbao
SPAIN

Abstract .- *This paper investigates sufficient and necessary and sufficient conditions for sampling time reachability and observability in hybrid multirate sampling data linear systems for the cases of fast input and fast output sampling rates .*

1. Introduction

In the last years considerable attention has been paid to the research on discrete sampling systems. Such systems are important in practice because of the widespread use of digital computers for active control and the importance of the sampling rate in achieving suitable stabiliy degrees and transient performances [1 - 7] . Recent studies have extended the classical ones by specifically introducing modern mathematical tools like the L_2 - induced norm considered as an operator between square integrable and piecewise continuous signals to minimize the worst possible amplification factor between selected signals what mathematically corresponds to minimizing the L_2 - induced norm of a linear operator (see [2 - 6]) . In [2] and [3] , upper bounds for the induced norms of particular sampled - data operators are derived and used for stability and robustness analysis. The above mentioned induced norm has been considered as an operator between square integrable and piecewise continuous signals and then optimized for the sampled - data obtained by using a standard digital controller on an analog plant [6] . On the other hand , hybrid systems have been found very useful to formulate real processes , including situations of adaptive and multirate sampling with essentially manipulate both kinds of signals [9 - 11] . In this paper , the reachability properties of multirate sampling hybrid linear systems are investigated in terms of formulating necessary and sufficient conditions for such properties to hold. Such systems are characterized by the feature that both continuous signals appear jointly with discrete signals sampled at different rates . This allows the accommodation of the various sampling rate requirements in terms of sampling efficiency , bandwidth or noise filtering at different points of the control loop. The paper is organized as follows . Section 2 gives some results about sufficient , and necessary and sufficient , conditions for reachability in discrete hybrid systems which are then used to derive extensions to the multirate case which is focused in section 3 . Two typical cases are considered , namely , the case when the state - output is sampled faster than the input and that when the input is sampled faster . For expository abbreviation, it is considered that all the components of each of the involved vectors are sampled at the same rate . Extensions to the case when the various components of the continuous and discrete subsystems could be sampled at different rates are pointed out in the simple case when there are two discrete subsystems one being sampled at a double sampling period . The formulation of the observability problem in hybrid multirate systems is briefly described in section 4 by giving extensions from the previous reachability results , finally , conclusions end the paper .The proofs of the results are omitted due to space reasons .

2. The hybrid state - space model

2.1 Hybrid continuous - discrete model

Consider the linear finite dimensional sampled - data system with hybrid state space representation of [6] become

$$\begin{bmatrix} \dot{x}_c(t) \\ x_d[k+1] \end{bmatrix} = \begin{bmatrix} A_c + A_{cs}(t)S_k & A_{cd}(t) \\ A_{ds}S_k & A_d \end{bmatrix} \begin{bmatrix} x_c(t) \\ x_d[k] \end{bmatrix} + \begin{bmatrix} B_c + B_{cs}(t)S_k \\ B_d S_k \end{bmatrix} u(t) \quad (2.1)$$

$$y(t) = \begin{bmatrix} B_c + B_{cs}(t)S_k & C_d(t) \end{bmatrix} \begin{bmatrix} x_c(t) \\ x_d[k] \end{bmatrix} + [D_c + D_d(t)S_k]u(t) \quad (2.2)$$

with T being the sampling period , $kT \leq t \prec (k+1)T$, where $x_c(t) \in \mathbf{R}^{n_c}$ is the analog state variable , $x_d[k] \in \mathbf{R}^{n_d}$ is the digital state variable , S_k is the sampling operator with sampling period T satisfying $(S_k v)(t) = v(kT)$, $u(t) \in \mathbf{R}^m$ is the piecewise continuous input , $y(t) \in \mathbf{R}^p$ is the output , A_c , A_{ds} , A_d , B_c, B_d , C_c and D_c are constant matrices of appropriate dimensions , and $A_{cs}(t)$, $A_{cd}(t)$, $B_{cs}(t)$, $C_{cs}(t)$, $C_d(t)$, $D_d(t)$ are T - periodic matrices of appropriate dimensions . At the sampling points $t = kT$, (2 .1) - (2 .2) become

$$x[(k+1)T] = Ax(kT) + Bu(kT) \quad (2.3)$$

$$y(kT) = Cx(kT) + Du(kT) \quad (2.4)$$

where $A = A_1(T)$, $B = B_1(T)$, $C = C_1(T)$, $D = D_1(T)$, and

$x(kT) = \left[x_c^T(kT) \quad x_d^T[k] \right]^T$, $u[k] = u(kT) = S_k u(t)$ $y[k] = y(kT) = S_k y(t)$ and

$$A_1(T) = \begin{bmatrix} e^{A_c T} + \int_0^T e^{A_c(T-\tau)} A_{cs}(\tau)\, d\tau & \int_0^T e^{A_c(T-\tau)} A_{cd}(\tau) d\tau \\ A_{ds} & A_d \end{bmatrix}$$

$$B_1(T) = \begin{bmatrix} \int_0^T e^{A_c(T-\tau)} B_{cs}(\tau) d\tau & \int_0^T e^{A_c(T-\tau)} B_c B_c^T e^{A_c^T(T-\tau)} d\tau \\ B_d & 0 \end{bmatrix} \quad (2.5)$$

$$C_1(T) = [C_c + C_{cs}(T), C_d(T)] \quad , \quad D_1(T) = [D_c + D_d(T)]$$

since the time - varying matrices in (2 . 1) - (2 . 2) are T - periodic . The next result establishes tests for Sampling- Time Reachability (STR) , [6] , (i . e. , sample time reachability from arbitrary initial conditions under piecewise continuous control on [0 , kT] for some sampling instant kT, of any prefixed state $x(kT) = (x_c^T(kT), x_d^T[k])^T = (x_c^{0\,T}, x_d^{0\,T})^T$ of the system (2. 3) - (2.5) . Its proof is based on the application of the Banach 's Perturbation Lemma , [12] , under reachability of the (A , B) pair , to a square complex matrix which is nonsingular irrespective of the complex argument under guaranteed reachability of the current system from the Popov - Belevitch - Hautus reachability text .

2. 2 Preliminary results

Lemma 2.1 . - The next propositions hold :

(i) The system (2 .3) - (2 .5) is STR iff rank [z I - A , B] = $n_c + n_d$ for all complex number z ε sp (A) (with sp (A) denoting the spectrum of the A - matrix) [Popov -Belevitch - Hautus rank reachability test , [8]] . This property implies and is implied by the nonsingularity of appropriate $n_c x n_c$ and $n_d x n_d$ polynomial matrices at the eigenvalues of A .

(i i) The system (2.3) - (2 . 5) is STR iff any of the two next conditions , which are equivalent to the rank condition given in (i) , hold : (a) the reachability matrix $R = [B, AB ,..., A^{s-1}B]$ has rank $n = n_c + n_d$ for some nonnegaive integer s $\leq n$. (b) The reachability grammian $\sum_{j=0}^{s} A^j B B^T A^{jT}$ is nonsingular .

(i ii) Let system (2. 3) - (2. 5) satisfy the next assumptions :

Assumptions 1 : (a) The pairs ($e^{A_c T}$, $\int_0^T e^{A_c(T-\tau)} B_c$) and (A_d , [A_{ds} , B_d]) are both reachable , and

(b) $\| \tilde{P}_{11} \|_2 \leq \epsilon_1 \prec \| P_{11s} \|_2$, $\| P_{12s} \|_2 \prec \epsilon_2$, $\| \tilde{P}_{12} \|_2 \prec \epsilon_3$

where $\tilde{P}_{11} = P_{11} - P_{11s}$, $\tilde{P}_{12} = P_{12} - P_{12s}$ $P_{22} = P_{22s}$ (the subscripts s denoting the static parts of the matrices , namely , those which do not involve time - varying marices) , and

$$P_{11s} = (zI - e^{A_c T})(zI - e^{A_c^T T}) + (\int_0^T e^{A_c(T-\tau)} B_c B_c^T e^{A_c^T (T-\tau)} d\tau)^2 \quad , \quad P_{12s} = -(zI - e^{A_c T}) A_{ds}^T \qquad (2.6)$$

for all z being an eigenvalue of A , ϵ_i (i = 1 , 2, 3) being some positive real constants and $\| (.) \|_2$ being the l_2 - matrix norm . Then , the system (2. 3) to (2. 5) is STR provided that

$$\epsilon_1 \prec \frac{1}{\| P_{22} \|_2} (\| P_{11s} \|_2 \| P_{22} \|_2 - 2(\epsilon_2^2 + \epsilon_3^2))$$

$$\epsilon_2^2 + \epsilon_3^2 \prec \frac{1}{2} \| P_{11s} \|_2 \| P_{22} \|_2 \qquad (2.7)$$

for all z ε sp (A) .

(iv) Proposition (iii) also holds if Assumption 1 (b) changes to a choice of the samplig period fulfilling $0 \prec T \prec \text{Min}_{\alpha \succ 4(\sigma+2\varepsilon)} \left\{ \frac{1}{3\varepsilon^2}(\alpha - 4(\sigma+2\varepsilon)) \ , \ \frac{\vartheta + \varepsilon' - 2\varepsilon(\sigma+\varepsilon)}{\alpha \varepsilon^2} \right\}$

where ϑ , σ , ε and ε' are real constants satisfying the constraints $\vartheta + \varepsilon' \succ 2\varepsilon(\sigma + \varepsilon)$,

$\sigma \geq \text{Max}(|z| : z \varepsilon \text{ sp}(A))$, $\vartheta = \text{Min}(\| zI - A_d \|^2 + \| zI - e^{A_c T} \|^2 : z \varepsilon \text{sp}(A))$

$\varepsilon \geq \text{Max}_{0 \leq \tau \leq t} (\| e^{A_c(T-\tau)} \| \ , \ \| A_{cs}(\tau) \| \ , \ \| A_{cd}(\tau) \| \ , \ \| B_{cs}(\tau) \| \ , \ \| A_d \| \ , \ \| A_{ds} \| \ , \ \| B_d \|)$

$$\varepsilon' = \left\| \int_0^T e^{A_c(T-\tau)} B_c B_c^T e^{A_c^T(T-\tau)} d\tau + A_{ds} A_{ds}^T + B_d B_d^T \right\| \qquad (2.8)$$

(v) Proposition (iii) also holds if Assumption 1(a) changes to the (A , B) - pair being reachable and Assumption 1(b) is changed to the next constraint on the sampling period

$$0 < T < \text{Min}_{\alpha > 4\sigma + 6\varepsilon} \left\{ \frac{1}{3\varepsilon^2}(\alpha - 4(\sigma + 2\varepsilon) \quad , \quad \frac{\vartheta + \varepsilon' - 2\varepsilon(\sigma + \varepsilon)}{\alpha \varepsilon^2}) \right\} \qquad (2.9)$$

***

Remark 2 . 1 . - The reachability of the hybrid system (2.1) is ensured under specific reachability conditions for the continuous and discretized subsystems appearing implicitly in (2. 1) - (2 . 2) of respective dimensions n_c and n_d with some extra companion conditions established in terms of the relative smallness of the norms of the interconnection matrices for such systems and the sampling period with respect to a measure of the sizes of the assumed reachable subsystems . In particular , in the case when the pairs (A_c , B_c) and (A_d , B_d) are both reachable , the time - dependent matrices in (2.1) are sufficiently small , with such a smallness being precisely characterizable , and the sampling period belongs to appropriate intervals so that the reachability is not lost under discretization (See Lemma 2.1 (iii) - (v)) .

***

The results of this section are extended in the sequel to some kinds of hybrid multirate systems , namely , systems that possess two of more different sampling rates affecting the state , input and output

3. Reachability of the multirate hybrid system

Now , multirate hybrid sampled - data systems extended from section 2 (i . e . , the input and output are sampled , in general , at different sampling rates) are considered. In the sequel , a sampling instant is called a simultaneous sampling instant if both the state / output and input are sampled at such a point. The base sampling period is the smaller sampling period in the loop which is that used for the parametrizations of the state - space description . The time - varying matrices in (2.1) - (2.2) are assumed to be periodic at the base sampling period . For convenience in the mathematical manipulations of the hybrid state equations , the state is assumed to be sampled at the base period so that all the remaining periods are multiple of the base one . Alternatively , a submultiple of the output sampling rate could be used for a fictitious discretization of the state being valid to describe the system. This stategy would be particularly useful when the various output components would be sampled at different sampling rates. Two cases are of specific interest for the subsequent study , namely :

a) *Case 1* (Fast output sampling) . All the components of the state and or output and input vectors are sampled at sampling rates T_0 (base period) and T_i , respectively , with $T_i = N T_0$, with N being a positive integer . The time - varying matrices in (2.1) - (2. 2) are assumed to be T_0 - periodic .

b) *Case 2* (Fast input sampling) . In this case , $T_0 = N T_i$ with T_i being the base period . The time - varying matrices in (2.1) - (2. 2) are assumed to be T_i - periodic .

***

3.1 Reachability conditions

Note that eqn. (2. 1) takes now the form

$$\begin{bmatrix} \dot{x}_c(t) \\ x_d[k+1] \end{bmatrix} = \begin{bmatrix} A_c + A_{cs}(t) S_k^0 & A_{cd}(t) \\ A_{ds} S_k^0 & A_d \end{bmatrix} \begin{bmatrix} x_c(t) \\ x_d[(k+1)N-1] \end{bmatrix} + \begin{bmatrix} B_c + B_{cs}(t) S_k^i \\ B_d S_k^i \end{bmatrix} u(t) \quad (3.1)$$

$$y(t) = \begin{bmatrix} B_c + B_{cs}(t) S_k^0 & C_d(t) \end{bmatrix} \begin{bmatrix} x_c(t) \\ x_d[(k+1)N] \end{bmatrix} + \begin{bmatrix} D_c + D_d(t) S_k^i \end{bmatrix} u(t) \quad (3.2)$$

with the digital , and analog state variables and the input and ouput being defined as in section 2 and $kT_0 \leq t < (k+1)T_0$ where $S^0{}_k$ and $S^i{}_k$ are the state - output and input sampling operator s with sampling periods T_0 and T_i satisfying ($S^0{}_k$) $v(t) = v(kT_0)$ and ($S^i{}_k$) $v(t) = v(kT_i)$, respectively , and the various matrices are defined as in (2. 1) - (2. 2) and have the same periodicity properties.

Case 1 . Eqns. (2. 1) - (2. 2) lead to the next recursive extended equations at the (state - output) sampling points $t = kT_0$:

$$x[(k+1)NT_0] = A\,x[kNT_0] + Bu[kNT_0] \quad (3.3)$$

$$y[kNT_0] = C\,x[kNT_0] + D\,u[kNT_0] \quad (3.4)$$

with the matrix redefinitions

$A = A_1(T_0)$, $B = B_1(T_0)$, $C = C_1(T_0)$, $D = D_1(T_0)$ according to (2 . 5) and

$x(kT_0) = [x_c^T(kT_0) \;\; x_d^T[k]]^T$, $u[k] = u(kT_0) = S_k^0 u(t)$

$y[k] = y(kT_0) = S_k^0 y(t)$ and

Case 2 : Eqns. (3.3) - (3. 4) still remain valid with all the matrices parametrized at T_i .

The next definition , established in terms of the input sampling rate and the sampling ratio , precises the concept of reachability in hybrid multirate sampling systems for Cases 1 -2 .

Definitions 3 . 1 . - The state x^0 is Sample Time Fast Output - Sampling Reachable (STFOSR) with pair (T_i, N) , i . e . , with input and output sampling rates T_i and $T_0 = T_i / N$, respectively , if there is a nonnegative finite integer 1 , nonncessarily being a multiple of N) and a sequence $u_{(.)}$ defined on [0 , 1 T_i / N] , such that the system (3. 3) - (3. 4) achieves any prefixed final state $x((k+1N)T_0) = x^0 = (x_c{}^0, x_d{}^0)^T$ from any $x(kT)$. The system is STFOSR with pair (T_i, N) iff all the sates are STFOSR. The definitions of the state x^0 , and the system (3. 3) - (3. 4) , being Sample Time Fast Input - Sampling Reachable (STFISR) with pair (T_i, N) , i . e . , with T_0 and $T_i = T_0 / N$, can be established ' mutatis - mutandis ' over the interval [0 , 1 T_0] = [0 , 1 N T_i] by using the parametrizations of (3. 4) - (3 .5) at T_i .

***

The next result is concerned with reachability conditions in the sense of Definition 3. 1 for the multirate system (3. 3) - (3.4) . Its proof is based on building the state trajectory under the time interval of interest and the subsequent use of the Rouché - Froebenius test related to solvability of algebraic linear equations and contradiction arguments to a set of equations which are built involving the reachability matrix .

Theorem 3 . 1 .- The next propositions hold :

(i) The system (3. 3) - (3.4) subject to (2.5) and parameterized at $T = T_0$ is STFOSR with (T_i , N) iff rank (R_{Nq}) = $n_c + n_d$ for some positive integer $q \geq \frac{(n_c + n_d)}{N} \geq \mu$ (degree of the minimal polynomial of $\tilde{A}$) , where the reachability matrix R_{Nq} for simultaneous sampling instants is

$$R_{Nq} = [\sum_{j=0}^{N-1} A^{N-j-1} B , \ldots , \sum_{j=0}^{N-1} A^{(q-1)N-j-1} B , \sum_{j=0}^{N-1} A^{qN-j-1} B] = [\tilde{B}, \tilde{A}\tilde{B}, \ldots, \tilde{A}^{q-1}\tilde{B}] \quad (3.5)$$

with $\tilde{A} = A^N = A_1^N(T_0)$, $\tilde{B} = A^N (\sum_{j=0}^{N-1} A^{-j-1}) B$. If the system is reachable at a simultaneous sampling instant $t = N q$ then it is also reachable at any state / output (simultaneous or no simultaneous) sampling instant $t = (qN + q')T_0$ for all nonnegative integer $q' < N$ with reachability matrix given by

$$R = [\sum_{j=0}^{q'-1} A^{q'-j-1} B , \sum_{j=0}^{N-1} A^{N+q'-j-1} B , \ldots , \sum_{j=0}^{N-1} A^{(q-1)N+q'-j-1} B] \quad (3.6)$$

where $l = (q-1)N + q' \prec qN$ with l and q being positive integers and q ' being a nonnegative integer .

(ii) The system (3. 3) - (3 . 4) subject to (2 .5) and parameterized at T_i is STFISR iff the reachability matrix ($R_{qN+q'}$) is full rank for some nonnegative integer q and some positive integer $q' < N$, where

$$R_{qN+j} = [\tilde{B} , \tilde{A}\tilde{B} , \ldots , \tilde{A}^{q-1}\tilde{B} , \tilde{A}^{q}\tilde{B}_j] \quad (3.7)$$

where $\tilde{A} = A_1(T_0) = A_1(NT_i)$, $\tilde{B} = \tilde{B}_N = [B' , A' B' , \ldots , A'^{N-1} B']$

with the reparametrizations

$$A' = \begin{bmatrix} e^{A_c T_i} [I + \int_0^{T_i} e^{-A_c \tau} A_{cs}(\tau) d\tau] & e^{A_c T_i} \int_0^{T_i} e^{-A_c \tau} A_{cd}(\tau) d\tau \\ 0 & I_{n_d} \end{bmatrix}$$

$$B' = \begin{bmatrix} \int_0^{T_i} e^{A_c (T_i - \tau)} (B_c + B_{cs}(\tau)) d\tau \\ B_d \end{bmatrix} , \quad \tilde{B}_j = [B' , A' B' , \ldots , A'^{j-1} B'] \quad (3.8)$$

for the given $q' < N$ and the $A_1(.)$ - matrix being defined in (2 . 1) . If the system is reachable then it is reachable at simultaneous sampling instants (i . e . , for some positive q and $q' = 0$) . ***

Note that , because of the structures of the reachability matrices , in order to check the reachability conditions in Theorem 3 . 1 , it suffices to consider q equal to the degree of the minimal polynomial of $A_1(T_0)$ and $\tilde{A}=A_1(NT_i)$, respectively. If such a rank test fails then the system is unreachable since the extra possibly added columns are linearly dependent on the former ones.

The next result is a direct application of Lemma 2. 1 to Theorem 3 .1 .

Corollary 3 . 1 .- The next propositions hold :

(i) The the system is STFOSR (respectively , STFISR) - reachability of the system (3. 3) - (3. 4) , subject to the parametrizations (2. 5) with $T = T_0$ can be tested through the nonsingularity of appropriate $n_c \times n_c$ and $n_d \times n_d$ polynomial matrices at the eigenvalues of $A = \tilde{A}(T_0)$ (respectively , $A = A(T_i)$) . Those matrices are defined as in Lemma 2 . 1 (i) , eqns . (3. 5) (respectively , (3.7)) .

($\|\tilde{P}_{11}\|_2 \leq \epsilon_1 \prec \|P_{11s}\|_2$, $\|P_{12s}\|_2 \prec \epsilon_2$, $\|\tilde{P}_{12}\|_2 \prec \epsilon_3$ i i) The system (3 . 3) - (3. 4) is STFOSR (respectively , STFISR) , subjected to the parametrizations (2. 5) with $T = T_i$ iff any of the three next conditions hold : (a) The reachability matrix , eqn. (3 . 5) (respectively , eqn. (3. 7)) has rank $n_c + n_d$ for any positive integer $q \geq \mu$ (degree of $\tilde{A}$) (respectively , $\tilde{A}'$) ; (b) For the above q , the reachability grammian $\sum_{j=0}^{q} \tilde{A}^j \tilde{B} \tilde{B}^T \tilde{A}^{jT}$ is nonsingular ; (c) [Popov - Belevitch - Hautus rank test [8]] rank [zI - $\tilde{A}$, $\tilde{B}$] = $n_c + n_d$ for all eigenvalue z of A .

(i ii) Let system (3. 3) - (3. 4) satisfy :

Assumptions 2 : (a) The pairs $(e^{A_c T_i}, \int_0^{T_i} e^{A_c(T_i-\tau)} B_c)$ and $(A_d, [A_{ds}, B_d])$ are both reachable

and (b) $\|\tilde{P}_{11}\|_2 \prec \epsilon_1 \prec \|P_{11s}\|_2$, $\|P_{12s}\|_2 \prec \epsilon_2$, $\|\tilde{P}_{12}\|_2 \prec \epsilon_3$

with the constants $\epsilon_{(.)}$ satisfying the constraints

$$\epsilon_1 \prec \frac{1}{\|P_{22}\|_2} \left(\|P_{11s}\|_2 \|P_{22}\|_2 - 2(\epsilon_2^2 + \epsilon_3^2) \right)$$

$$\epsilon_2^2 + \epsilon_3^2 \prec \frac{1}{2} \|P_{11s}\|_2 \|P_{22}\|_2 \qquad (3.9)$$

for all $z \in sp(A)$ from Lemma 2.1 , with the parametrizations (2. 5) at $T_i = N T_0$, where

$\tilde{P}_{11} = P_{11} - P_{11s}$, $\tilde{P}_{12} = P_{12} - P_{12s}$, and

$$P_{11s} = (zI - e^{A_c T_i})(zI - e^{A_c^T T_i}) + \left(\int_0^{T_i} e^{A_c (T_i - \tau)} B_c B_c^T e^{A_c^T (T_i - \tau)} d\tau\right)^2 \quad , \quad P_{12s} = -(zI - e^{A_c T_i}) A_{ds}^T \qquad (3.10)$$

for all z being an eigenvalue of A , $\varepsilon_{(.)}$ being some positive real constants and $\|(.)\|_2$ being the l_2 - matrix norm . Then , the system (3. 3) - (3. 4) is STFOSR . [Similar conditions canbe established for the system to be STFISR] . ***

3 . 2 The case of component - by - component different sampling rates

Now assume that the various state and input and output components of the vectors describing system (2.1) - (2. 2) can be sampled at different sampling rates . In this situation , the above formulation can be easily extended . To simplify the exposition , assume that there are only two different state sampling rates T_{s1} and T_{s2} and define $T = l . c . m .$ (least common multiple) (T_{s1} , T_{s2}). Assume that $T_{s1} = T_{s2} / 2$ so that there are two discrete substates x_{d1} and x_{d2} in (2.1) - (2.2) . Assume also that there are two input sampling rate periods corresponding to both discrete subsystems $T_{i1} = T_{s1}$, and $T_{i2} = T_{s2} = T_{i1} / 2$ with $T' = T = l . c . m$ (T_{i1} , T_{i2}) $= 2 T_{i1}$. Assume that the continuous substate is also sampled at rate $T' / 2$. Thus , if k is a nonnegative integer governing the sampling instants at period T , eqns. (2. 1) - (2 . 2) become

$$\begin{bmatrix} \dot{x}_c(t) \\ x_{d1}[kT + T/2] \\ x_{d2}[(k+1)T] \end{bmatrix} = \begin{bmatrix} A_c + A_{cs}(t) S_{0k}(0) & A_{cd1}(t) & A_{cd2}(t) \\ A_{ds1}(t) S_{0k}(0) & A_{d11} & A_{d12} \\ A_{ds2}(t) S_{0k}(0) & A_{d21} & A_{d22} \end{bmatrix} \begin{bmatrix} x_c(t) \\ x_{d1}[k] \\ x_{d2}[k] \end{bmatrix} + \begin{bmatrix} B_c + B_{cs}(t) S_k^c \\ B_{d1} S_{ik}^1(0) \\ B_{d2} S_{ik}^2(0) \end{bmatrix} u(t) \qquad (3.11)$$

where $(S_{0k}^1(j))v(t) = v(kT + jT/2), j = 0,1 \; ; (S_{0k}^2)v(t) = v(kT) , t \varepsilon [kT, (k+1)T]$ and , similarly , the hold operators acting on the input S_{ik}^c and $S_{ik}^{j(.)}$ could be defined . In this particular case , S_k^c and $S_{ik}^{1(.)}$ are identical but generalizations to the general case of multiple input and state sampling rates are immediate at the expense of more involved expressions in (3. 11) . The state - space description is now calculated by taking two consecutive samples for the first discrete substate while one sample for the second substate since $T_{s2} = 2 T_{s1}$. This leads directly to

$$\begin{bmatrix} \dot{x}_c(t) \\ x_{d1}[k+1] \\ x_{d2}[k+1] \end{bmatrix} = \begin{bmatrix} A_c + A_{cs}(t) S_{0k}(1) + A_{cd1}(t) A_{ds1} S_{0k}(0) & A_{cd1}(t) A_{d11} & A_{cd1}(t) A_{d12} + A_{cd2}(t) \\ A_{ds1}(t) S_{0k}(1) + A_{d11}(t) A_{ds1} S_{0k}(0) & A^2_{d11} & A_{d11} A_{d12} + A_{d12} \\ A_{ds2}(t) S_{0k}(1) + A_{d21} S_{0k}(0) & A_{d21} A_{d11} & A_{d21} A_{d12} + A_{d22} \end{bmatrix}$$

$$\begin{bmatrix} x_c(t) \\ x_{d1}[k] \\ x_{d2}[k] \end{bmatrix} + \begin{bmatrix} A_{cd1}(t) + B_{d1} S^c{}_{ik}(0) + B_c(t) + B_{cs}(t) S^c{}_{ik}(1) \\ A_{d11}(t) + B_{d11} S^1{}_{ik}(0) + B_{d1} S^1{}_{ik}(1) \\ B_{d2} S^2{}_{ik} \end{bmatrix} u(t) \qquad (3.12)$$

for all $0 \le t \le T$. Now , the reachability problem is solved by calculating the state transition over a period $T = T'$ from (3. 12). Extensions to the general case of multiple sampling rates for the input , state and output components is completely similar although requiring more involved calculations .

4. Remarks on the observability of the multirate hybrid system

First , consider the discrettized state - space description of the multirate sampling system (3. 3) - (3.4) , subjected to the parametrization (2.5) at the sampling rate T = g. c . m (T_{0j} , j = 1 , 2 , ... , δ) for some integer $\delta \leq p$. The $T_{0(.)}$ are the various sampling rates each used for an individual (or a group of) output component (s) . If $\delta = p$ then all the output components are sampled at different sampling rates . An extended observability definition from [8] will be introduced in the sequel to extend the above reachability results to the observability of a class of multirate sampling systems .

Definition 4 . 1 . - For the system (3. 3) - (3 .4) , the vector $[x_c^{0T}, x_d^{0T}]^T$ is unobseavable if , using as initial condition $[x_c^{0T}, \bar{x}_d^{0T}]^T$, $u(t) = o \Rightarrow y(t) = 0$ for almost all $t \geq 0$. The system (3.3) - (3.4) , subject to the above parametization is observable iff the only unobservable state is the origin . ***

The assumption of free s ystem (i. e . , u (t) = 0 for all time) used in Definition 4. 1 is unessential to the subsequent discussion about the observability but facilitates the mathematical development s . The next result is concerned with the observability of the given class of multirate sampling system . Its proof is performed by defining the relevant observability grammian over the time interval of interest and the evaluation of the output energy over such an interval by using that grammian and the application of Definition 4 . 1 .

Theorem 4 . 1 . - The multirate sampling system of Definition 4 . 1 is observable iff the pair ($A = A_1(T)$, C) is observable where

$$C^T C = \int_0^T [C_1(\tau) \ , \ C_2(\tau)]^T C_1(\tau) \ , \ C_2(\tau)] \, d\tau \qquad (4.1)$$

which is positive semidefinite with

$$C_1(\tau) = C_c(e^{A_c \tau} + \int_0^\tau e^{A_c(\tau-\tau')} A_{cs}(\tau') d\tau') + C_{cs}(\tau)$$

$$C_2(\tau) = C_c \int_0^\tau e^{A_c(\tau-\tau')} A_{cd}(\tau') d\tau' + C_d(\tau) \ ; \ 0 \leq \tau \leq T \qquad (4.2)$$

***

Definition 4. 1 and Theorem 4 . 1 motivate the next result which also implicitly includes an equivalent definition for observability .

Corollary 4 . 1 . - The system (3. 3) - (3. 4) , subject to the parametrization (2. 5) at T is observable iff rank [C^T , $A^T C^T$, ... , $(A^T)^k C^T$] = $n_c + n_d$ for $k \geq \mu - 1$ (μ being the degree of the minimal polynomial of (A)) . The least - squares solution for x_0 is $\hat{x}_0 = (OO^T)^{-1} O[y^T(0), y^T(1), ..., y^T(\mu-1)]^T$ or , equvalently , rank [$zI - A^T$, C^T] = $n_c + n_d$ for all complex $z \varepsilon$ sp (A).

Remark 4 . 1 . - Note that the nonsingularity of M(k) in Theorem 4 . 1 and Corollary 4 .1 can be ensured from the Perturbation Banach ' s Lemma if $M_s(k)$ is nosingular and $\|\tilde{M}(k)\| < \|M_s(k)\|$ for any choice of the nonsingular matrix $M_s(k)$. Thus , $M_s(k)$ and $\tilde{M}(k)$ can be associated with with a static / nominal observability matrix in the same context as the reachability matrices were manipulated in Theorem 3 . 1) . Therefore , different sufficient conditions of observability of the current system can be obtained from that one being interpreted as the nominal hybrid system and sufficient smallness in matrix norms of the disturbed observability matrix which is associated with the terms not being included in the observability matrix of the nominal system . Such a nominal system can be time - invariant , if suitable by design or analysis conveniences , in the case when the time - varying matrices are small in norm by proceeding in a similar way as in the reachability case. (See Theorem 3 . 1 for the reachability case).

5. Conclusions

This paper has dealt with the investigation of the properties of reachability and observability of a class of hybrid multirate sampling systems where the inputs , outputs and states are sampled at different rates . Sufficient and / or necessary and sufficient conditions guaranteeing such properties have been proved . The obtained results are based on assumptions of reachability (observabilty) of the static parts of the individual subsystems (namely , the continuous and discrete subsystems beeing free of time - varying parameters) that constitute the hybrid system as well as on the smallness of either the sampling periods or the system ' s parameters .

ACKNOWLEDGMENTS

The author is very grateful to DGICYT (Project PB93-0005) and UPV by their support of this work .

References

[1] G. F. Franklin and J. D. Powell , Digital Control of Dynamic Systems . New York : Addison - Wesley , 1980.
[2] P. M . Thompson , G. Stein , and M . Athans , " Conic sectors for sampled data feedback systems , " Syst. Contr. Lett. , Vol. 3, pp. 77 - 82 , 1983.
[3] P. M. Thompson , R. L. Dailey , and J. C . Doyle , " New conic sectors for sampled data feedback systems , " Syst. Contr. Lett. , Vol. 7 , pp. 395 - 404, 1986.
[4] B. A Francis and T. T. Georgiu , " Stability theory for linear time invariant plants with periodic digital controllers , " IEEE Trans. Automat. Contr., Vol. 33, pp. 820 - 832, 1988.
[5] G. M. Leung , T.P. Perry , and B. A. Francis , " H 2 optimal sampled data control , " IEEE Trans. Automat. Contr. , Vol. 36 , pp. 387 - 397 , 1991.
[6] P. T. Kabamba and S. Hara , " Worst - case analysis and design of sampled - data control systems , " IEEE Trans. Automat. Contr. , Vol. 38 , No. 9 , pp. 1337 - 1357 , 1993.
[7] M. de la Sen , " A method for improving the adaptation transients using adaptive sampling, " Int. J. of Control , Vol. 40 , No. 4 , pp. 639 - 665, 1984.
[8] T. Kailath , Linear Systems . Englewood Cliffs , N. J. : Prentice - Hall , Inc. , 1980.
[9] D. P. Glasson and J. Dowd , Research multirate estimation and control - optimal sample rate selection , Technical Report TR - 1356 - 2 , The Analytic Science Corporation , Reading , MA, 1981.
[10] M. de la Sen , " Multirate digital adaptive control " , Comput. and Mathematics with Applc. , Vol. 11, No. 12, pp. 1193 - 1210 , 1985.
[11] M. de la Sen , "Multirate hybrid adaptive control " , IEEE Trans. Automat. Contr. , Vol. AC - 31 , pp. 582 - 586 , 1986.
[12] J. M. Ortega , Numerical Analysis . New York : Academic Press , 1972.

Robust Stability of a Polytope of Matrices: A Global Linear Programming Solution†

B. R. Barmish*, C. A. Floudas**, C. V. Hollot*** and R. Tempo****

Abstract

In this paper, a global solution to some open robustness problems is given. The starting point is the known fact that a number of "unresolved" robustness issues reduce to stability questions about a polytope of matrices. Therefore, we begin with a matrix having entries which depend affinely on uncertain parameters; these parameters are restricted to a box. Subsequently, we provide a solution to the robust stability problem. This solution involves only a finite number of linear programs. More specifically, using the problem data, we define a finitely terminating sequential procedure for linear program generation with each new set of LPs dependent on the solution of the preceding set. Associated with this finite sequence of LPs, we also provide a handle on computational effort in the following sense: Given any prescribed accuracy level ϵ for the global optimum, we guarantee its attainment via solution of 2n problems where n is the dimension of the matrix under consideration. In turn, each of these problems can be solved in a finite number of stages N_ϵ each involving no more than $4n - 2$ linear programs. This, however, does not imply polynomial complexity because the dependence of N_ϵ on n is unknown.

*ECE Department, University of Wisconsin, Madison, WI 53706. **Department of Chemical Engineering, Princeton University, Princeton, NJ 08544. ***ECE Department, University of Massachusetts, Amherst, MA 01003. ****CENS-CNR, Politecnico di Torino, Torino, Italy. † Funding for this research was provided by the National Science Foundation under Grants ECS-9111570, NSF-8858366, CBT-8857013, CTS-9221411 and CENS-CNR of Italy.

1. Introduction

Over the last several years, some of the more difficult robust stability problems (those not admitting analytical solutions) have been attacked via mathematical programming. In making a distinction between a strong result and a weak result, two critical issues are *global provability* and *computational complexity.* That is, it is important to know whether the mathematical program to be solved lends itself to a globally optimal solution involving a "reasonable" amount of computation. In this regard, note that an algorithm which only provides convergence to a local optimum amounts to no more than a sufficient condition for robustness. For example, suppose one establishes that robust stability is guaranteed if and only if the minimum value, call it f^*, of some mathematical program is less that unity. Then clearly, the calculation of some local minimum does not necessarily indicate whether $f^* < 1$. For the case when a global solution *can* be guaranteed, the issue of computational complexity becomes increasingly important as the dimension of the problem data increases.

With the emphasis on global provability, we see a number of results in the literature which rely heavily on convexity; e.g., see [1] – [3]. That is, for a classical convex program, we are assured that every local minimum is a global minimum. The main limitation of the convex programming approach is that the class of uncertainty structures to which it applies is quite restrictive; e.g., systems with affine linear dependence on uncertain parameters or systems in the μ setting with three or less uncertainty blocks; e.g., see [4] and [5].

To deal with more general uncertainty structures, there are a number of papers in the literature which involve iterative methods associated with a mathematical programming reformulation of the underlying robustness problem; e.g., see [6] – [8]. In each case, global provability can be established but computational complexity is a major difficulty. Branching operations are required and one cannot guarantee the desirable property that the algorithm will execute in polynomial time.

This paper also falls into the line of research aimed at finding a computationally efficient mathematical programming solution. However, there is one distinguishing feature which differentiates this work from previous results — the central role of linear programming in our approach. We do not simply reformulate the robustness problem as a mathematical program with nonlinear objective or constraints. Instead, we heavily exploit the structure of the robustness problem in order to expose some fundamental linear programming problems at the core. We prove that there is a rich class of robustness problems which lend themselves to global solution via a finite sequence of linear programs.

The key to the method in this paper is the construction of the "right" optimization problem — one for which a global linear programming solution is possible. To this end, a sequence of three papers [9] - [11] in the field of global optimization have influenced our approach. In particular, these papers concentrate on a class of nonconvex programs with objective and constraint functions satisfying some mild technical conditions plus an important biconvexity requirement. The proofs of the results of this paper are reported in [12].

2. Robust Stability of a Polytope of Matrices

The objective of this section is to provide a precise formulation of the classical robust stability problem under consideration. We also point out a number of important special cases which are covered by the theory. Indeed, we begin with an $n \times n$ real matrix A whose entries are affine linear functions of some vector

$$q \doteq (q_1, q_1, \ldots, q_\ell)$$

of real *uncertain parameters* which typically represent physical quantities whose values are imprecisely known. We write

$$A(q) \doteq A_0 + \sum_{i=1}^{\ell} q_i A_i$$

with A_0 assumed stable and each of the A_i above being a fixed $n \times n$ matrix. We also refer to the entries $a_{i,j}(q)$ of $A(q)$.

The problem formulation also includes interval bounds for each uncertain parameter q_i. Hence, with Q denoting the rectangular uncertainty set in $\mathbf{R}^{\ell}$, we focus attention on the robust stability problem for the *polytope of real matrices*

$$\mathcal{A} \doteq \{A(q) : q \in Q\}.$$

More precisely, we say that $\mathcal{A}$ is *robustly stable* if, for every $q \in Q$, the matrix $A(q)$ has all its eigenvalues in the strict left half plane.

3. Preliminary Technical Results: Reformulation

In this section, we concentrate on preliminary technical manipulations needed in order to ultimately reduce the polytope stability problem to linear programming.

3.1 Beginning of Reformulation: To get around the fundamental difficulty associated with expansion of the determinant above, the first step involves exploitation of a specially constructed $2n \times 2n$ matrix which has previously appeared in the literature in an unstructured perturbation context; see [13].

3.1.1 Lemma: *Let I denote the $n \times n$ identity matrix and define the $2n \times 2n$ matrix*

$$A^{+}(\omega, q) \doteq \begin{bmatrix} A(q) & \omega I \\ -\omega I & A(q) \end{bmatrix}$$

with entries $a^{+}_{i,j}(\omega, q)$. Then, $\mathcal{A}$ is robustly stable if and only if

$$A^{+}(\omega, q)v \neq 0$$

for all $\omega \geq 0$, $q \in Q$ *and nonzero* $v \in \mathbf{R}^{2n}$.

3.2 An Intermediate Nonconvex Program: Our objective in this subsection is to create a nonconvex minimization problem whose global convergence properties will be studied in the sequel. This result should be viewed as temporary in the sense that we plan to eliminate the nonconvexity.

In the the next two lemmas, we continue with the reformulation of the robust stability problem. To this end, we obtain a frequency bound

$$\Omega \doteq [0, \omega_{max}]$$

and consider the problem of finding

$$f^* \doteq \min\{||A^+(\omega, q)v|| : \omega \in \Omega;\ q \in Q;\ ||v|| = 1\}$$

with arbitrary norms for $A^+(\omega, q)v$ and v above.

3.2.1 Lemma: *There exists a distinguished frequency* $\omega_{max} > 0$ *having the following property: If* $\omega > \omega_{max}$, *then* $\lambda = j\omega$ *is not an eigenvalue of* $A(q)$ *for all* $q \in Q$.

3.2.2 Lemma: *The polytope of matrices* $\mathcal{A}$ *is robustly stable if and only if*

$$f^* > 0.$$

3.3 Convexification of Domain: To motivate the next part of the reformulation, we note that the optimization problem associated with Lemma 3.2.2 has a nonconvex constraint set for v. Namely, the constraint $||v|| = 1$ in the definition of f^* defines a nonconvex set for v and precludes use of many standard mathematical programming methods. The lemma to follow overcomes this difficulty; important to the proof is the fact that positivity of f^* in Lemma 3.2.2 is invariant to the choice of norms in the associated optimization problem.

In the lemma below, there are two critical technical elements. First, the proof exploits flexibility in the choice of norm for v. Second, we partition the v space into $2n$ convex polytopes (diamonds) and show that it suffices to solve a separate optimization problem for each.

3.3.1 Lemma: *For each* $i^* \in \{1, 2, \ldots, 2n\}$, *let*

$$\mathcal{V}_{i^*} \doteq \{v \in \mathbf{R}^{2n} : v_{i^*} = 1;\ \sum_{i \neq i^*} |v_i| \leq 2n - 1\}$$

and

$$f_{i^*} \doteq \min\{||A^+(\omega, q)v|| : \omega \in \Omega;\ q \in Q;\ v \in \mathcal{V}_{i^*}\}.$$

Then, $\mathcal{A}$ *is robustly stable if and only if*

$$f_{i^*} > 0$$

for all $i^* \in \{1, 2, \ldots, 2n\}$.

3.4 Reformulation Using ℓ^∞ Norm: To complete the reformulation of the robust stability problem, we now exploit the flexibility in the choice of norm for $A^+(\omega, q)v$. Indeed, since robust stability of $\mathcal{A}$ is invariant to the choice of norm for $A^+(\omega, q)v$ in Lemma 3.3.1, we use the standard ℓ^∞ norm. This eventually leads us to an optimization problem which is solvable via linear programming. To begin, we observe that

$$||A^+(\omega, q)v||_\infty = \max_{u \in \mathcal{U}} u^T A^+(\omega, q)v$$

where

$$\mathcal{U} \doteq \{u \in \mathbf{R}^{2n} : \sum_{i=1}^{2n} |u_i| \leq 1\}$$

is the unit ball in ℓ^1 norm. Subsequently, the following lemma is essential.

Step 1a (Lower Bound): For $k \geq 1$, find (ω^k, q^k, t^k) achieving the *lower bound*

$$LB_k = \min\ t$$

subject to $\omega \in \Omega, q \in Q$ and

$$t \geq \mathcal{L}(\omega, q, u^\ell);\ \ell = 0, 1, \ldots, k-1.$$

Step 1b: (Upper Bound): With (ω^k, q^k) generated above, let

$$u^k \in \arg\max_{u \in \mathcal{U}}\ \mathcal{L}(\omega^k, q^k, u).$$

Then, the *upper bound* is given by

$$UB_k = \mathcal{L}(\omega^k, q^k, u^k).$$

Step 2 (Termination): Replace k by $k+1$ and check if the termination condition

$$UB_k - LB_k < \epsilon$$

is satisfied. If not, return to Step 1.

4.2 The Main Result: We are now prepared to state the main result in this paper.

4.2.1 Theorem: *For fixed index $i^* \in \{1, 2, \ldots, 2n\}$, consider the optimization problem in Lemma 3.3.1 associated with robust stability of $\mathcal{A}$. Then, using the sequence $(\omega^k, q^k, u^k, t^k)$ generated via Steps 0 - 2 above, the optimality bound*

$$LB_k \leq f_{i^*} \leq UB_k$$

is guaranteed for all k. Moreover, the termination condition

$$UB_k - LB_k < \epsilon$$

is satisfied in a finite number of steps.

3.4.1 Lemma: *For fixed* $i^* \in \{1, 2, \ldots, 2n\}$, *it follows that* f_{i^*} *is given by the minimum value of the mathematical program*

$$\min \ t$$

subject to linear inequality constraints

$$\omega \in \Omega, \quad q \in Q$$

and bilinear constraints

$$t \geq \mathcal{L}(\omega, q, u); \ \ u \in \mathcal{U}$$

where

$$\mathcal{L}(\omega, q, u) = \sum_{i=1}^{2n} u_i a^+_{i,i^*}(\omega, q) - (2n-1) \max_{j \neq i^*} |\sum_{i=1}^{2n} u_i a^+_{i,j}(\omega, q)|.$$

4. Main Result: The Linear Programming Solution

This section is devoted to exposition of the main result, Theorem 4.2.1. To this end, we consider the optimization problem formulated in Lemma 3.3.1. In view of the lemma, we henceforth work with a fixed index $i^* \in \{1, 2, \ldots, 2n\}$.

4.1 Finite Sequence of Linear Programs: We now proceed to describe the finite sequence of linear programs which is used to solve the robust stability problem. We view the algorithm as "high level" in the following sense: Each of the steps below requires a degree of elaboration in order to be executed via an LP code. For example, when absolute values of variables occur, there are standard LP "tricks" for rewriting constraints as linear inequalities.

Step 0 (Initialization): An error tolerance $\epsilon > 0$ for f_{i^*} is specified and we select any starting point $(\omega, q) = (\omega^0, q^0) \in \Omega \times Q$. Subsequently, we use linear programming to obtain

$$u^0 \in \arg\max_{u \in \mathcal{U}} \mathcal{L}(\omega^0, q^0, u).$$

References

[1] S. P. Boyd and C. H. Barrat, *Linear Controller Design: Limits of Performance*, Prentice Hall, Englewood Cliffs, NJ, 1991.

[2] Yu. Nesterov and A. Nemirovskii, *Interior Point Polynomial Methods in Convex Programming: Theory and Applications*, SIAM, Philadelphia, PA, 1993.

[3] S. P. Boyd, L. El Ghaoui, E. Feron and V. Balakrishnan, *Linear Matrix Inequalities in Systems and Control Theory*, SIAM Publications, Philadelphia, PA, 1994.

[4] J. Doyle, "Analysis of Feedback Systems with Structured Uncertainty," *IEE Proceedings*, vol. 129, part D, pp. 242-250, 1982.

[5] M. G. Safonov, "Stability Margins of Diagonally Perturbed Multivariable Systems," *IEE Proceedings*, vol. 129, part D, pp. 251-256, 1982.

[6] R. R. de Gaston and M. G. Safonov, "Exact Calculation of the Multiloop Stability Margin," *IEEE Transactions on Automatic Control*, vol. AC-33, pp. 156–171, 1988.

[7] A. Sideris and R. S. Sanchez Pena, "Fast Computation of the Multivariable Stability Margin for Real Interrelated Uncertain Parameters," *IEEE Transactions on Automatic Control*, vol. AC-34, pp. 1272-1276, 1989.

[8] A. Vicino, A. Tesi and M. Milanese, "Computation of Nonconservative Stability Perturbation Bounds for Systems with Nonlinearly Correlated Uncertainties," *IEEE Transactions on Automatic Control*, vol. AC-35, pp. 835-841, 1990.

[9] J. F. Benders, "Partitioning Procedures for Solving Mixed-Variables Programming Problems," *Numerische Mathematik*, vol. 4, pp. 238-252, 1962.

[10] A. M. Geoffrion, "Generalized Benders Decomposition," *Journal of Optimization Theory and Applications*, vol. 10, pp. 237-260, 1972.

[11] C. A. Floudas and V. Visweswaran, "Primal-Relaxed Dual Global Optimization Approach," *Journal of Optimization Theory and Applications*, vol. 78, pp. 187-225, 1993.

[12] B. R. Barmish, C. A. Floudas, C. V. Hollot and R. Tempo, "A Global Linear Programming Solution to Some Open Robustness Problems Including Matrix Polytope Stability," *University of Wisconsin-Madison Technical Report* ECE-94-13, 1994.

[13] L. Qiu and E. J. Davison, "The Stability Robustness Determination of State Space Models with Real Unstructured Perturbations," *Mathematics of Control Signals and Systems*, vol. 4, pp. 247-267, 1991.

The Permutation Trellis Complexity of Linear Block Codes*

Sam Dolinar, Laura Ekroot, Aaron Kiely, Wei Lin, and Robert J. McEliece

Department of Electrical Engineering and Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California, USA

Abstract.

Let $\mathbb{C}$ be a fixed (n, k) linear block code, and let $E(\mathbb{C})$ denote the trellis complexity of $\mathbb{C}$, which we take to be the number of edges in the Wolf trellis for $\mathbb{C}$. In this paper we consider the problem of finding the minimum value of $E(\pi\mathbb{C})$, where π ranges over all possible coordinate permutations of $\mathbb{C}$. We denote this minimum value by $E^*(\mathbb{C})$, and call it the permutation trellis complexity of $\mathbb{C}$. Using ideas previously introduced by Forney and others, together with some new ones of our own, we develop a series of analytical lower bounds on $E^*(\mathbb{C})$, which depend on knowing the minimum distance, and sometimes the dual minimum distance, of $\mathbb{C}$. We also present a heuristic algorithm for finding "good" coordinate permutations for a given code $\mathbb{C}$. We conclude with a table which compares the lower bounds with the upper bounds obtained from the heuristic algorithm. In many cases, including the $(24, 12, 8)$ Golay code, and the $(48, 24, 12)$ quadratic residue code, we find that the upper and lower bounds agree; for these codes the permutation trellis complexity is then known exactly.

---

* The work of Dolinar, Ekroot, and Kiely, and a portion of the work of McEliece, was done at the Jet Propulsion Laboratory, California Institute of Technology, under contract to the National Aeronautics and Space Administration. Wei Lin's work, and part of McEliece's work, was supported by AFOSR grant no. F4960-94-1-005. A portion of McEliece's work was also supported by a grant from Pacific Bell.

1. Introduction and Summary.

In 1978 Wolf [24] presented, for the first time, a method of representing the words in a linear block code by the path labels in a trellis, thus uncovering an important connection between block and convolutional codes, and in particular showing that the Viterbi algorithm can be used to decode an arbitrary block code. Later that same year Massey [16] made a further study of the problem of representing a block code by a trellis, and gave an alternative construction. For the next 10 years, there was relatively little work in this area, but in 1988 Forney [6], in a now celebrated appendix to a paper on coset codes, described what he called "the trellis diagram of a code," which resulted in an explosion of interest in the subject. Of the post-Forney work, perhaps the most noteworthy is that of Muder [20], who showed that among all trellises representing a given block code, the Forney trellis minimized the number of vertices at each depth. For this reason, Muder called the Forney trellis the "minimal" trellis for the code, and the name has stuck. Since then, there have been many papers dealing with the minimal trellis for a given block code or class of block codes, e.g., [2], [3], [9], [10], [11], [13], [21], and [23].

In a recent paper, [19], McEliece studied the computational complexity of a generalized version of the Viterbi algorithm on a trellis, and concluded that this complexity is proportional to the number of edges in the trellis. Motivated by this result, he then showed that among all trellises representing a fixed linear block code $\mathbb{C}$, the trellis introduced by Wolf in 1978 uniquely minimizes the edge count. He also showed that the Wolf trellis is isomorphic to the Forney-Muder "minimal trellis," an historically important fact overlooked by Forney and Muder, but announced by Kot and Leung [11] in 1993. McEliece also observed that among all trellises for $\mathbb{C}$, the Wolf trellis also minimizes the vertex count, the maximum vertex dimension, and, among a large but restricted class of trellises for $\mathbb{C}$, the number of trellis "bifurcations." Thus the Wolf trellis is definitively the uniquely optimal trellis structure for $\mathbb{C}$.

If the code $\mathbb{C}$ is being used on a memoryless channel, then permuting the columns of $\mathbb{C}$ will not affect the code's performance, but as Massey [16] was the first to note, a column permutation of $\mathbb{C}$ may drastically alter its trellis complexity. Thus if Σ_n represents the group of all $n!$ permutations of the set $\{1, 2, \ldots, n\}$, and if for $\pi \in \Sigma_n$ we denote by $\pi\mathbb{C}$ the code obtained by applying the permutation π to the coordinates of $\mathbb{C}$, the main goal of this paper is to study the quantity $E^*(\mathbb{C})$, defined as

$$E^*(\mathbb{C}) = \min_{\pi \in \Sigma_n} E(\pi\mathbb{C}), \tag{1.1}$$

where $E(\pi\mathbb{C})$ denotes the number of edges in the Wolf trellis for $\pi\mathbb{C}$. We call the quantity $E^*(\mathbb{C})$ *the permutation trellis complexity* of the code $\mathbb{C}$. Other authors, e.g., [2], [3], [9], [10], [21], and [23] have all considered the "permutation trellis complexity" of linear codes, but with no standardization of the measure of complexity. It is our hope that, based on the results of [19], researchers in this area will focus on the problem of computing $E^*(\mathbb{C})$. As a start in this direction, we have included in this paper, in Section 4, a short table of some interesting binary block codes and the best known upper an lower bounds on their permutation trellis complexity. (An expanded discussion of permutation complexity, which considers other complexity measures, e.g., vertex count, bifurcation count, maximum vertex degree, span, appears in [12], which is currently in preparation.)

Here is a brief outline of the rest of the paper. In Section 2, we will derive a number of lower bounds on $E^*(\mathbb{C})$, based on ideas of several other authors, as well as some of our own. In Section 3, we will describe a family of heuristic algorithms that, as a rule, find "good" permutations for a given code $\mathbb{C}$, and hence also upper bounds on $E^*(\mathbb{C})$. Finally, in Section 4, we will give a table, listing the best upper and lower bounds for $E^*(\mathbb{C})$ that we know of for a number of interesting codes.

2. Lower Bounds on Permutation Trellis Complexity.

In this section, we will outline several methods for obtaining lower bounds on the number of edges in the Wolf trellis for any column-permutation of a given (n, k) binary linear block code $\mathbb{C}$. In the interests of brevity, we will omit some details, which can be found in [7], [12], or [19].

We begin with a simple formula for the number of edges $E(\mathbb{C})$ in the Wolf trellis for a fixed (n, k) binary linear code $\mathbb{C}$, using the notation of [19].

$$E(\mathbb{C}) = \sum_{i=0}^{n-1} E_{i,i+1}(\mathbb{C}), \tag{2.1}$$

where $E_{i,i+1}$ denotes the number of edges joining vertices at depth i to those at depth $i+1$ in the trellis. The value of $E_{i,i+1}$ is given by the formula

$$E_{i,i+1}(\mathbb{C}) = 2^{k-p_i(\mathbb{C})-f_{i+1}(\mathbb{C})}. \tag{2.2}$$

where $p_i(\mathbb{C})$ and $f_i(\mathbb{C})$ are the dimensions of $\mathbb{C}$'s ith "past" and "future" subcodes $P_i(\mathbb{C})$ and $F_i(\mathbb{C})$, defined as follows.

$$P_i(\mathbb{C}) = \{\boldsymbol{C} \in \mathbb{C} : C_{i+1} = C_{i+2} = \cdots = C_n = 0\}, \tag{2.3}$$

$$F_i(\mathbb{C}) = \{\boldsymbol{C} \in \mathbb{C} : C_1 = C_2 = \cdots = C_i = 0\}. \tag{2.4}$$

The importance of the past and future subcodes was first discovered by Forney [6, Appendix A]. Combining eqs. (2.1) and (2.2), we obtain the following explicit formula for $E(\mathbb{C})$:

$$E(\mathbb{C}) = \sum_{i=0}^{n-1} 2^{k-p_i(\mathbb{C})-f_{i+1}(\mathbb{C})}. \tag{2.5}$$

In order to convert eq. (2.5) to a lower bound on $E^*(\mathbb{C})$, i.e., a lower bound on $E(\pi\mathbb{C})$ which is valid for all permutations π, we will find universal *upper* bounds for the corresponding quantities p_i and f_i that appear in eq. (2.5). To this end, let us define

$$K_i(\mathbb{C}) \stackrel{\text{def}}{=} \max_{\pi \in \Sigma_n} p_i(\pi\mathbb{C}). \tag{2.6}$$

From the definition eq. (2.3), it follows that $K_i(\mathbb{C})$ is the largest possible dimension of a subcode of $\mathbb{C}$ defined by requiring $n - i$ of the codeword coordinates to be 0. That

is, if $S \subseteq \{1, 2, \ldots, n\}$, and if we define the S-shortened subcode $\mathbb{C}_S$ to be the set of codewords of $\mathbb{C}$ which vanish on S, and $K_S(\mathbb{C})$ the corresponding dimension, then

$$K_i(\mathbb{C}) = \max\{K_S(\mathbb{C}) : |S| = n - i\}. \tag{2.7}$$

Similarly, since by eq. (2.4), the future subcode for $\mathbb{C}$ is obtained by shortening $\mathbb{C}$ on a specific set of i positions, it follows that

$$\max_{\pi \in \Sigma_n} f_i(\pi\mathbb{C}) = K_{n-i}(\mathbb{C}). \tag{2.8}.$$

The important sequence $K_0(\mathbb{C}), \ldots, K_n(\mathbb{C})$ has been called the *dimension/length profile* (DLP) of the code $\mathbb{C}$ by Forney [7]. In summary, then, for all $\pi \in \Sigma_n$,

$$p_i(\pi\mathbb{C}) \leq K_i(\mathbb{C}) \tag{2.9}$$

$$f_i(\pi\mathbb{C}) \leq K_{n-i}(\mathbb{C}). \tag{2.10}$$

Combining eqs. (2.5), (2.9), and (2.10), we obtain the lower bound

$$E^*(\mathbb{C}) \geq \sum_{i=0}^{n-1} 2^{k - K_i(\mathbb{C}) - K_{n-i-1}(\mathbb{C})}, \tag{2.11}$$

which we call *the DLP bound* on $E^*(\mathbb{C})$. As a rule, this bound is not easy to apply, since the DLP is hard to compute for most codes. (An important exception to this is that Kasami et al. [9] have determined the DLP for the entire family of Reed-Muller codes, and have further shown, in effect, that the DLP bound of eq. (2.11) holds with equality for every RM code.)

We can convert eq. (2.11) to a bound on $E^*(\mathbb{C})$ which is in general looser than (2.11), but is on the other hand easier to compute, as follows. Suppose that the minimum distance of $\mathbb{C}$ is d, and that the minimum distance of the dual code $\mathbb{C}^\perp$ is $d^\perp$. We indicate this by saying that $\mathbb{C}$ is an $(n, k, d, d^\perp)$ code. If we then define

$$\overline{K}_i(n, k, d, d^\perp) \stackrel{\text{def}}{=} \max\{K_i(\mathbb{C}) : \mathbb{C} \text{ is an } (n, k, d, d^\perp) \text{ code}\}, \tag{2.12}$$

we then have from eqs. (2.9) and (2.10),

$$p_i(\pi\mathbb{C}) \leq \overline{K}_i(n, k, d, d^\perp) \tag{2.13}$$

$$f_i(\pi\mathbb{C}) \leq \overline{K}_{n-i}(n, k, d, d^\perp), \tag{2.14}$$

and we also have, from eq. (2.11), that

$$E^*(\mathbb{C}) \geq \sum_{i=0}^{n-1} 2^{k - \overline{K}_i(n,k,d,d^\perp) - \overline{K}_{n-i-1}(n,k,d,d^\perp)}. \tag{2.15}$$

The bound in eq. (2.15) looks formidable, because very little is known directly about the numbers $\overline{K}_i(n, k, d, d^\perp)$, but good upper bounds on them can be obtained, as follows. First note that if $\mathbb{C}$ is an $(n, k, d, d^\perp)$ code, it is certainly also a (n, k, d) code, and so if we define $K(i, d)$ to be the largest possible dimension for a binary linear code of length

i and minimum distance d, it follows from eq. (2.12), and the fact that a subcode of an $(n, k, d, d^\perp)$ code shortened on $n - i$ positions is effectively an (i, d) code, that

$$\overline{K}_i(n, k, d, d^\perp) \leq K(i, d). \tag{2.16}$$

Combining eq. (2.16) with eq. (2.15), we obtain

$$E^*(\mathbb{C}) \geq \sum_{i=0}^{n-1} 2^{k-K(i,d)-K(n-i-1,d)}. \tag{2.17}$$

a bound which is useful when the dual distance $d^\perp$ is unknown. The numbers $K(i, d)$ can be calculated or estimated using any of the techniques described in [15, Chapter 17], or, if the code parameters are small, the required values can be simply looked up in useful tables of Brouwer and Verhoeff [5].

If the dual distance $d^\perp$ is known, we can further refine eq. (2.17). Since the dual code of any $(n, k, d, d^\perp)$ code is an $(n, n-k, d^\perp)$ code, and since Forney [7, Theorem 3], and Kiely et al. [12, Lemma 12] have shown that the DLP for the dual code $\mathbb{C}^\perp$ is related to the DLP for the primal code $\mathbb{C}$ by the formula

$$K_i(\mathbb{C}) = i - (n - k) + K_{n-i}(\mathbb{C}^\perp), \quad \text{for } i = 0, \ldots, n, \tag{2.18}$$

it follows that

$$\overline{K}_i(n, k, d, d^\perp) \leq i - (n - k) + K(n - i, d^\perp). \tag{2.19}$$

Thus combining eqs. (2.13), (2.14), (2.16), and (2.19), we have, for all permutations π, and for all $i - 0, 1, \ldots, n$, the important inequalities

$$p_i(\pi C) \leq \begin{cases} K(i, d) \\ i - (n - k) + K(n - i, d^\perp) \end{cases} \tag{2.20}$$

$$f_i(\pi C) \leq \begin{cases} K(n - i, d) \\ k - i + K(i, d^\perp) \end{cases} \tag{2.21}$$

Furthermore, if we define

$$K_i^*(n, k, d, d^\perp) \stackrel{\text{def}}{=} \min(K(i, d), i - (n - k) + K(n - i, d^\perp)), \tag{2.22}$$

we also have the lower bound

$$E^*(\mathbb{C}) \geq \sum_{i=0}^{n-1} 2^{k-K_i^*(n,k,d,d^\perp)-K_{n-i-1}^*(n,k,d,d^\perp)}. \tag{2.23}$$

Since the numbers $K_i^*(n, k, d, d^\perp)$ are simple combinations of the numbers $K(i, d)$, they too can be calculated or estimated using [15, Chapter 17] or [5].

In Section 4, we will give a table giving the values of the lower bound eq. (2.23) for a number of interesting codes. In the remainder of this section, we will illustrate the use of the techniques developed here in two Examples.

2.1 Example. Let C be the $(7,4,3)$ binary Hamming code. Its dual code $\mathbb{C}^\perp$ is a $(7,3,4)$ code. Using the bounds (2.20) and (2.21), we have

$$p_i(\pi\mathbb{C}) \le K(i,d)$$
$$f_i(\pi\mathbb{C}) \le K(n-i,d),$$

and using the tables of $K(i,d)$ given in [5], we find the following upper bounds on the past and future dimension of any permutation $\pi\mathbb{C}$:

i:	0	1	2	3	4	5	6	7
p_i:	0	0	0	1	1	2	3	4
f_i:	4	3	2	1	1	0	0	0.

This table, combined with eq. (2.11), gives $E^*(\mathbb{C}) \ge 32$. If there were a permutation $\pi\mathbb{C}$ attaining these values, then by the dual code relationships (see [7] or [12])

(2.24) $$f_i^\perp = n - k - i + p_i$$

(2.25) $$p_i^\perp = -k + i + f_i$$

the corresponding parameters of the dual code $\pi\mathbb{C}^\perp$ would be as given below.

i:	0	1	2	3	4	5	6	7
$p_i^\perp$:	0	0	0	0	1	1	2	3
$f_i^\perp$:	3	2	1	1	0	0	0	0,

which, by eq. (2.11), gives $E^*(\mathbb{C}^\perp) \ge 28$. But from $p_3 = 1$ and $d = 3$, we conclude that $\boldsymbol{x} = (111000)$ is a word in $\pi\mathbb{C}$, and from $p_4^\perp = 1$ we conclude that $\boldsymbol{y} = (1111000)$ is a word in $\pi\mathbb{C}^\perp$, but $\boldsymbol{x} \cdot \boldsymbol{y} = 1$, a contradiction. So the above "profiles" aren't achievable. Now from eqs. (2.24) and (2.25), we have $p_4^\perp = f_4$, and $f_3^\perp = p_3$. But as we have seen above, if $p_3 = 1$, then $p_4^\perp = 0$; and if $p_4^\perp = 1$, then $p_3 = 0$. Thus it follows, for any permutation π, if $p_3(\pi\mathbb{C}) = 1$, then $f_4(\pi\mathbb{C}) = p_4^\perp(\pi\mathbb{C}) = 0$, and if $f_4(\pi\mathbb{C}) = 1$, then $p_3(\pi\mathbb{C}) = f_3^\perp(\pi\mathbb{C}) = 0$. Thus the best profiles we can hope for any permutation of $\mathbb{C}$ are

i:	0	1	2	3	4	5	6	7
p_i:	0	0	0	1	1	2	3	4
f_i:	4	3	2	1	**0**	0	0	0

or

i:	0	1	2	3	4	5	6	7
p_i:	0	0	0	**0**	1	2	3	4
f_i:	4	3	2	1	1	0	0	0

where the boldface entries show the difference from the original values. Either of these tables, combined with the DLP bound (2.11), give $E^*(\mathbb{C}) \ge 36$. Similar reasoning about the dual code shows that the best possible profiles are

i:	0	1	2	3	4	5	6	7
$p_i^\perp$:	0	0	0	0	1	1	2	3
f_i:	3	2	1	**0**	0	0	0	0

or

i:	0	1	2	3	4	5	6	7
p_i:	0	0	0	0	**0**	1	2	3
f_i:	3	2	1	1	0	0	0	0

either of which gives $E^*(\mathbb{C}^\perp) \ge 32$. And indeed all of these profiles are achievable. For example, the first profile for $\mathbb{C}$ is achieved by the permutation with (minimal-span) generator matrix

$$G = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix},$$

and the second profile is obtained by reversing the order of the columns. For $\mathbb{C}^{\perp}$, a minimal-span generator matrix that achieves the first listed profile is

$$G^{\perp} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix},$$

and the second profile is obtained by reversing the order of the cloumns of $G^{\perp}$. Thus we conclude that for the $(7,4,3)$ Hamming code, $E^*(\mathbb{C}) = 36$ and $E^*(\mathbb{C}^{\perp}) = 32$. (See also Example 3.1.) ▪

2.2 Example. In this example we consider the $(48,24,12)$ extended quadratic residue code. Since this code is self-dual, we have $d^{\perp} = 12$ and so by (2.20) and (2.21), we have

$$p_i \le \begin{cases} K(i,12) & \text{for } 0 \le i \le 24 \\ i - 24 + K(48-i,12) & \text{for } 24 \le i \le 48 \end{cases}$$

$$f_i \le \begin{cases} 24 - i + K(i,12) & \text{for } 0 \le i \le 24 \\ K(48-i,12) & \text{for } 24 \le i \le 48. \end{cases}$$

If we look up the necessary values of $K(i,d)$ in the tables of [5], the above inequalities give the following upper bounds p_i^* and f_i^* on the $(p_i(\pi\mathbb{C}), f_i(\pi\mathbb{C}))$ profile or any permutation $\pi\mathbb{C}$ of the code.

i:	0	1	2	3	4	5	6	7	8	9	10	11	12
p_i^*:	0	0	0	0	0	0	0	0	0	0	0	0	1
f_i^*:	24	23	22	21	20	19	18	17	16	15	14	13	13

i:	13	14	15	16	17	18	19	20	21	22	23	24
p_i^*:	1	1	1	1	1	2	2	2	3	3	4	5
f_i^*:	12	11	10	9	8	8	7	6	6	5	5	5,

with the values of p_i^* and f_i^* with $25 \le i \le 48$ given by the formula $p_i^* = f_{48-i}^*$ and $f_i^* = p_{48-i}^*$. Using these upper bounds, and the formula in eq. (2.11), we find that $E^*(\mathbb{C}) \ge 860,156$ for this code. Surprisingly, with the aid of the heuristic algorithm described in the next section, we have been able to find a permutation of this code which achieves this optimum profile, and so $E^*(\mathbb{C}) = 860,156$ for the $(48,24,12)$ extended quadratic code. (See Example 3.3.) ▪

3. A Heuristic Algorithm for Finding Good Column Permutations.

In Section 2 we developed a number of lower bounds on the permutation-trellis complexity of a given block code. In this section, we will describe a heuristic algorithm we have developed which as a rule succeeds in finding a permutation with relatively low trellis complexity. The objective function used by the algorithm is the *span* of the code, rather than the edge count. Let us briefly review the definition of the span of a code here. (For details, refer to [19].)

If $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ is a nonzero binary n-vector, its *left index*, denoted $L(\boldsymbol{x})$, is the smallest index i such that $x_i \neq 0$. Similarly, the *right index* of $\boldsymbol{x}$, denoted $R(\boldsymbol{x})$ is the largest index i such that $x_i \neq 0$. The *span* of $\boldsymbol{x}$, denoted $\mathrm{Span}(\boldsymbol{x})$, is the discrete "interval" $[s,t] = \{L(\boldsymbol{x}), L(\boldsymbol{x})+1, \ldots, R(\boldsymbol{x})\}$. The *spanlength* of $\boldsymbol{x}$, denoted $\mathrm{spanlength}(\boldsymbol{x})$, is the number of elements in $\mathrm{Span}(\boldsymbol{x})$, i.e., $\mathrm{spanlength}(\boldsymbol{x}) = |\mathrm{Span}(\boldsymbol{x})|$.

If $G = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k)$ is a $k \times n$ binary matrix, with rows $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$, its *spanlength*, denoted $\mathrm{spanlength}(G)$, is the sum of the spanlengths of the rows. A *minimal span* generator matrix for G is one for which the spanlength is a minimum, and this minimum value is called (with slight abuse of notation) the *span* of the code. In [19] it is shown that an arbitrary generator matrix for a code $\mathbb{C}$ can be converted into a MS generator matrix by a simple algorithm that uses only elementary row operations. In [19] many equivalent definitions of a MSGM are given, but for us it suffices to cite only one: a generator matrix G is a MSGM if and only if the left indices of its rows are distinct, and the right indices of the rows are also distinct (the "LR Property" of [19].)

The span is much easier to compute than the edge count, but there is a close connection between low edge count and low span. (In fact, we do not know of a code for which a permutation that minimizes the span does not also minimize the edge count, but we doubt if this is a general truth.)

The algorithm consists of three parts: the *preprocessor*, the *permuter*, and the *postprocessor*. The preprocessor consists of preliminary row operations, which put G into either *RRE* or *MS* form, and then sort the rows so that the weight of row 1 is a minimum, the weight of row 2 in those positions where row 1 is zero is a minimum, the weight of row 3, in those positions where rows 1 and 2 are both zero is a minimum, etc. A generator matrix with this form is denoted G_1. If the columns of G_1 are viewed as k-bit binary expansions of integers in the range 0 to $2^k - 1$, then k-tuples, then the "column permuter" arranges the columns in ascending order, with the resulting matrix denoted G_2. Finally, the postprocessor converts the matrix G_2 into MS form, G_3, so that the span can be computed.

3.1 Example. As in Example 2.1, let's start with the cyclic $(7,4,3)$ Hamming code with generator polynomial $g(x) = x^3 + x + 1$, and corresponding "cyclic" generator matrix

$$G_1 = \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \end{array},$$

which, having the LR property, is minimal-span (span = 16), and is also already in

"G_1" form. The decimal equivalents of the seven columns are as follows:

column	1	2	3	4	5	6	7
decimal equivalent	8	4	10	13	6	3	1

Thus, sorting the columns into increasing order gives the matrix

$$G_2 = \begin{array}{c} \begin{matrix} 7 & 6 & 2 & 5 & 1 & 3 & 4 \end{matrix} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}.$$

Using the "minimal-span" algorithm given in [19, Section 6], we obtain the following minimal-span form for G_2:

$$G_3 = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

The span of G_3 is seen to be 15, one less than that of G_1, and indeed this is the minimum possible span among all column-equivalent versions of the original code C, and the corresponding trellis has 36 edges, which is also a minimum, as we saw in Example 2.1. ▪

3.2 Example. The algorithm as just described may not find the best column permutation. For example, for the $(5, 3, 2)$ code with RRE generator matrix

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix},$$

if we sort the rows according to the description above, we obtain

$$G_1 = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

If we then reorder the columns into ascending order, we obtain

$$G_2 = \begin{array}{c} \begin{matrix} 3 & 1 & 4 & 2 & 5 \end{matrix} \\ \begin{pmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix} \end{array}.$$

The MS form of G_2 is

$$G_3 = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix},$$

which has span 8, and edge count 16, which is not the smallest possible. To overcome this kind of problem, we can modify the algorithm to allow preliminary pairwise column exchanges. For example, if we interchange columns 1 and 4 of the original matrix G, and then apply the algorithm we arrive at

$$G_3' = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix},$$

which has smaller span (7) and edge count (14) than the previous permutation, and indeed this permutation is globally minimal.

3.3 Example. We have applied our heuristic algorithm to the extended $(48, 24, 12)$ quadratic residue code, and have found a permutation that achieves the "(p_i, f_i) profile lower bound" we calculated in Example 2.2. For reference, we give a minimal-span generator matrix for a permutation that achieves this profile:

$$\begin{bmatrix}
000000000000000000000000000000000000111111111111 \\
000000000000000000000000000000111111111111000000 \\
000000000000000000000000000111000111111000111000 \\
000000000000000000000000001101100100110010011010 \\
000000000000000000000000110010011010101001101100 \\
000000000000000000000011101110101010110000100000 \\
000000000000000000001100110010101000111100000000 \\
000000000000000000010101100011100100011001010000 \\
000000000000000001100001010110001001001111000000 \\
000000000000000110010000010110101110101000000000 \\
000000000000001010010110011010001001110000000000 \\
000000000000011110111110000001100001000000000000 \\
000000000000100001000001111110011110000000000000 \\
000000000011010000100010111111001000000000000000 \\
000000000110110000101011000011100000000000000000 \\
000000001001110111011011001011101000000000000000 \\
000000010111000101001110111000000000000000000000 \\
000000101110100000101100110101000000000000000000 \\
000001100000101000101011101110000000000000000000 \\
000011011000110110101011000000000000000000000000 \\
000111000111001110100110000000000000000000000000 \\
001110110100011100111000000000000000000000000000 \\
010101110001111111000000000000000000000000000000 \\
111111111111000000000000000000000000000000000000
\end{bmatrix}$$

■

The algorithm described in this section seems to work quite well for codes of even length, but for odd length codes, its performance is sometimes disappointing. We have discovered a useful trick that overcomes this problem in many cases. If we have an odd-length (n, k) code $\mathbb{C}$, we apply the algorithm to the extended, even length, $(n+1, k)$ code $\mathbb{C}'$, obtaining a good permutation for $\mathbb{C}'$. We then delete the overall parity-column from the permuted, extended code. If the extended code is invariant under a transitive permutation group, then any column of the extended code can be deleted [15, Corollary

8.15], so that by deleting the column which reduces the span as much as possible, a good permutation for the original code usually results.

In the final section of the paper, we will give the results of applying our algorithms to a number of interesting codes.

4. Tables of Trellis Complexity.

In this section, we present a table listing some interesting binary block codes, together with our current best estimate of their permutation trellis complexities. In the table, the "upper bound" represents the fewest edges in any Wolf trellis for a permutation of the listed code that our algorithm has been able to find. The "VDC" is the Viterbi decoding complexity, which is the number of edges in the best trellis found so far, divided by the number of information bits k in the code. This is a rough measure of decoding complexity per decoded bit, if the Viterbi algorithm is used on the trellis, and it allows a fair comparison between codes with different n's, k's, and d's, and indeed between block and convolutional codes. For example, the standard $K = 7$ rate $1/2$ convolutional code has 128 edges per trellis segment, and each edge is labelled with two bits, so that the "VDC" for this code is $128 \cdot 2 = 256$ edges per decoded bit. The "Lower Bound" is the bound from eq. (2.23).

Code	Upper Bound	VDC	Lower Bound	Comments
[7,4,3,4] *Hamming code*	36	9	32	36 is optimal (Example 2.1)
[8,4,4,4] *Self–dual code*	44	11	44	Optimal
[15,5,7,4] *BCH code*	156	31.2	148	
[15,7,5,4] *BCH code*	300	42.8	172	
[15,11,3,8] *BCH code*	196	17.8	148	
[16,5,8,4] *BCH code*	172	34.4	172	Optimal
[16,7,6,4] *BCH code*	420	60	236	
[16,11,4,8] *BCH code*	252	22.9	252	Optimal
[17,9,5,6] *QR code*	412	45.8	380	Ref. [23]
[18,9,6,6] *QR code*	732	81.33	508	
[23,12,7,8] *Golay code*	3068	255.7	2684	
[24,12,8,8] *Golay code*	3580	298.3	3580	Optimal
[31,10,12,5] *BCH code*	7068	706.8	3484	

Code	Upper Bound	VDC	Lower Bound	Comments
[31,10,12,5] *JMG code*	7500	750	3484	[4, Table 16.1]
[31,16,7,8] *BCH code*	5884	367.75	4796	
[32,6,16,4] *BCH code*	684	114	684	Optimal
[32,11,12,6] *BCH code*	11804	1073.09	5948	
[32,16,8,8] *BCH code*	6396	399.75	6396	Optimal
[32,21,6,12] *BCH code*	17340	825.7	8316	
[32,26,4,16] *BCH code*	1180	45.4	1180	Optimal
[47,24,11,12] *QR code*	794620	33109.2	645116	
[48,24,12,12] *QR code*	860156	35839.8	860156	(Example 2.2)
[64,39,10,8] *BCH code*	95839228	$2.46\ 10^6$	161020	
[64,45,8,16] *BCH code*	1739004	38644.5	143868	
[64,51,6,24] *BCH code*	231228	4533.88	82556	
[64,57,4,32] *BCH code*	5676	99.6	5084	

References.

1. Y. Be'ery and J. Snyders, "A recursive Hadamard transform optimal soft decision decoding algorithm," *SIAM J. Alg. Disc. Meth.*, vol. 8 (1987), pp. 778–789.

2. Y. Berger and Y. Be'ery, "Bounds on the trellis size of linear block codes." *IEEE Trans. Inform. Theory, vol. IT-39 (1993)*, pp. 764–773.

3. Y. Berger and Y. Be'ery, "Soft trellis-based decoder for linear block codes ." *IEEE Trans. Inform. Theory, vol. IT-40 (1994)*, pp. 203–209.

4. E. R. Berlekamp, *Algebraic Coding Theory*. New York: McGraw-Hill, 1968.

5. A. E. Brouwer and T. Verhoeff, "An updated table of minimum-distance bounds for binary linear codes." *IEEE Trans. Inform. Theory*, vol. 39 (1993), pp. 662–677.

6. G. D. Forney, Jr., "Coset codes — Part II: Binary lattices and related codes." *IEEE Trans. Inform. Theory, vol. IT-34 (1988), pp. 1152–1187.*

7. G. D. Forney, Jr., "On dimension/length profiles and trellis complexity of linear block codes," submitted to *IEEE Tran. Inform. Theory.*

8. G. D. Forney, Jr., and M. D. Trott, "The dynamics of group codes: state spaces, trellis diagrams, and canonical encoders." *IEEE Trans. Inform. Theory, vol. IT-39 (1993), pp. 1491–1513.*

9. T. Kasami, T. Takata, T. Fuijwara, and S. Lin, "On the optimum bit orders with respect to the state complexity of trellis diagrams for binary linear codes." *IEEE Trans. Inform. Theory, vol. IT-39 (1993), pp. 242–245.*

10. T. Kasami, T. Takata, T. Fuijwara, and S. Lin, "On the complexity of trellis structure of linear block codes." *IEEE Trans. Inform. Theory, vol. IT-39 (1993), pp. 1057–1064.*

11. A. D. Kot and C. Leung, "On the construction and dimensionality of linear block code trellises." *Proc. 1993 ISIT, p. 291.*

12. A. B. Kiely, S. Dolinar, R. J. McEliece, L. Ekroot, and W. Lin, "Trellis Decoding Complexity of Linear Block Codes," *in preparation.*

13. F. R. Kschischang and V. Sorokine, "On the trellis structure of block codes," submitted to *IEEE Trans. Inform. Theory.*

14. J. H. van Lint, *Coding Theory*. Berlin: Springer-Verlag, Lecture Notes in Mathematics no. 210, 1971.

15. F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam: North-Holland, 1977.

16. J. L. Massey, "Foundations and methods of channel coding." *Proc. Int. Conf. Inform. Theory and Systems, NTG-Fachberichte vol. 65 (1978), pp. 148–157.*

17. R. J. McEliece, *The Theory of Information and Coding*. Reading, Mass.: Addison-Wesley, 1977.

18. R. J. McEliece, "The Viterbi decoding complexity of linear block codes," *Proc. 1994 ISIT, p. 341.*

19. R. J. McEliece, "On the Wolf Trellis for Linear Block Codes," submitted to *IEEE Trans. Inform. Theory.*

20. D. J. Muder, "Minimal trellises for block codes." *IEEE Trans. Inform. Theory, vol. IT-43 (1988), pp. 1049–1053.*

21. A. Vardy and Y. Be'ery, "Maximum-likelihood soft decision decoding of BCH codes." *IEEE Trans. Inform. Theory. vol. IT-40 (March 1994), pp. 546–554.*

22. A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm." *IEEE Trans. Inform. Theory*, vol. IT-13 (April 1967), pp. 260–269.

23. Y.-Y. Wang and C.-C. Lu, "The trellis complexity of equivalent binary [17, 9] quadratic residue code is five." *Proc. 1993 ISIT, p. 200.*

24. J. K. Wolf, "Efficient maximum likelihood decoding of linear block codes," *IEEE Trans. Inform. Theory*, vol. IT-24 (1978), pp. 76–80.

A Heuristic for Ordering a Linear Block Code to Minimize Trellis State Complexity

Frank R. Kschischang and **Gavin B. Horn**
Department of Electrical and Computer Engineering
University of Toronto

Abstract — **An important problem in the theory and application of block code trellises is to find a coordinate permutation for a given code that minimizes the trellis state complexity. In this paper we show that the problem of minimizing a given component of the state complexity profile is NP-complete. We describe an algorithm, though not guaranteed to find an *optimal* coordinate ordering, uses a heuristic descent technique to find "good" solutions to the problem. We have applied this algorithm to various codes, and our results are tabulated.**

1 Introduction

Although the codes obtained by permuting the coordinates of a linear block code are equivalent, it is well known that the minimal trellises for these equivalent codes in general are not equivalent [1, 2, 3, 4, 5, 6, 7]; in particular, different coordinate permutations may yield trellises with different state complexity profiles. As J. L. Massey pointed out some time ago [2]: "the art of trellis decoding of a linear code [is] that of re-arranging the order of digits in the code word to obtain a non-systematic code for which [the trellis complexity] is as small as possible."

In this paper we show that a related problem, that of minimizing the ith component of the state complexity profile, is a hard problem in the same sense that many well-known problems in computer science (like the travelling salesman problem) are hard: namely that the problem is NP-complete [8]. This leads us to conjecture that the problem of minimizing the maximum trellis complexity is also NP-complete, but we have so far been unable to prove this. It is widely believed that P != NP, i.e., that no algorithms that find solutions to NP-complete problems with a running time bounded by a polynomial function of the length of the problem input exist. Thus, optimal solutions to the coordinate ordering problem appear to be computationally infeasible.

However, as with many combinatorial optimization problems, heuristic search algorithms may be used to find *good* solutions. This paper describes one such algorithm. We have used this algorithm to search for efficient coordinate orderings for various classes of codes, and some of our results are tabulated in this paper. Among the results to be presented is an ordering of the (16,7,6) binary lexicode [9] that is *componentwise* optimal, i.e., the state complexity profile meets the DLP lower bound [7].

This paper is organized as follows. A brief description of trellises and their state space complexity is given in Section 2. In Section 3, we formulate the problem of finding an efficient coordinate ordering as the "MAXIMUM PARTITION RANK" problem. We also

define the related "iTH PARTITION RANK" and "PARTITION RANK VECTOR" problems, and show how these problems are inter-related. An NP-completeness proof for iTH PARTITION RANK is sketched in Appendix A.

In Section 4, we describe our search procedure, which uses a heuristic descent method from random initial points to descend to a local optimum. Results of applying this algorithm to various binary codes, including BCH, extended BCH, quadratic residue, and lexicographic codes, are tabulated in Section 5. Section 6 offers some concluding remarks.

2 State Complexity

Formal definitions for block code trellises are given, e.g., in [2, 10, 11]. Briefly, a trellis for a block code C of length n is an edge-labeled directed graph with a distinguished "root" vertex having in-degree zero and a distinguished "goal" vertex have out-degree zero, with the following properties:

1. all vertices can be reached from the root;
2. the goal can be reached from all vertices;
3. the number of edges traversed in passing from the root to the goal along *any* path is n; and
4. the set of n-tuples obtained by "reading off" the edge labels encountered in traversing all paths from the root to the goal is C.

It is useful to think of the number of edges needed to reach a given vertex from the root as the "time index" of that particular vertex. Vertices with the same time index are referred to as the "states" at that time index.

The minimal trellis for a linear block code C over $GF(q)$ can be constructed in a number of different (but equivalent) ways, e.g., [1, 2, 11]. For linear codes, the set of states in the minimal trellis at a given time index i can be characterized [3, 12] as a vector space over $GF(q)$. Denoting the dimension of this vector space by $s_i(C)$, this means that there are $q^{s_i(C)}$ states at time index i. The vector $(s_0(C), \ldots, s_n(C))$ is the state complexity profile of C [7]. Its maximum component is $s_{\max}(C) \triangleq \max_i\{s_i(C)\}$.

If C' is a code equivalent to C obtained by permuting the coordinates of C, then $s_{\max}(C') \neq s_{\max}(C)$ in general. For example, the (8,4) binary Reed-Muller code, obtained by extending the cyclic (7,4) Hamming code has a state space complexity profile $(0, 1, 2, 3, 4, 3, 2, 1, 0)$, with $s_{\max} = 4$, whereas, when expressed in the coordinate ordering obtained from the iterated squaring construction described in [3], the (8,4) code has state space complexity profile $(0, 1, 2, 3, 2, 3, 2, 1, 0)$, with $s_{\max} = 3$.

The minimum $s_{\max}(C)$ over all coordinate orderings is the *state complexity* $s(C)$ of C. The state complexity, $s(C)$ has been described by Muder [10] as a fundamental descriptive characteristic of the code C, and this parameter governs the complexity of a straightforward maximum-likelihood decoding of the code using a breadth-first (Viterbi algorithm) trellis-based search. The main problem addressed in this paper is to find a coordinate permutation π of a code C so that $s_{\max}(\pi C) = s(C)$.

3 The Partition Rank Problem

As described in [3, Appendix] and other papers (notably [12], which treats the problem in a general group-theoretic setting), the state complexity profile for a given linear code

can be obtained in terms of the dimensions of various subcodes and projections of that code.

Let C be a linear code of length n over a field F. At time index i, denote by P_{i-} : $C \rightarrow F^n$ the operator that maps an n-tuple $c = (c_1, c_2, \ldots, c_n)$ to $(c_1, \ldots, c_i, 0, \ldots, 0)$, i.e., the operator that zeroes out the "*future*" of c. Similarly, denote by P_{i+} the operator that maps c to $(0, \ldots, 0, c_{i+1}, \ldots, c_n)$, i.e., the operator that zeroes out the "*past*" of c. The *images* of C under the operation of these projections are denoted by $P_{i-}(C)$ and $P_{i+}(C)$, respectively. The *kernel* of $P_{i-} : C \rightarrow F^n$ is the so-called *future subcode* C_{i+}; similarly, the kernel of $P_{i+} : C \rightarrow F^n$ is the so-called *past subcode* C_{i-}. Clearly, C_{i+} is the set of all codewords that are all zero in the past, while C_{i-} is the set of all codewords that are all zero in the future.

Denote by $k(C)$ the dimension of a code (or vector space) C. The state complexity at time index i is given by [3]

$$s_i(C) = k(C) - k(C_{i-}) - k(C_{i+}).$$

However, one of the basic properties of linear vector space transformations relates the dimensions of the image and kernel of the transformation to the dimension of the space being transformed: if $T : V \rightarrow W$ is a linear transformation with range R_T and kernel N_T, then $k(V) = k(R_T) + k(N_T)$. In the context of the linear projection maps P_{i-} and P_{i+}, we see that

$$k(C) = k(P_{i-}(C)) + k(C_{i+}) = k(P_{i+}(C)) + k(C_{i-}).$$

(This is Lemma 1 of [7]). We can therefore write

$$s_i = k(P_{i-}(C)) + k(P_{i+}(C)) - k(C). \tag{1}$$

Given a generator matrix G for C, the terms on the right hand side of (1) are easily computed in terms of the ranks of submatrices of G. For $1 \leq i < n$, partition G into two submatrices,

$$G = [A_i | B_{n-i}] \tag{2}$$

where A_i consists of the first i columns of G and B_{n-i} consists of the last $n - i$ columns of G. Then (1) is equivalent to

$$s_i = r(A_i) + r(B_{n-i}) - r(G), \tag{3}$$

where $r(\cdot)$ denotes the rank of the enclosed matrix. Note that $r(G) = k$ is a constant for a given (n, k) code.

Equivalently, since the state complexity profile for the dual code $C^\perp$ of C is identical to that C [3], we can replace the generator matrix G in (2) and (3) with the parity check matrix H (which is the generator matrix for $C^\perp$).

For $1 \leq i < n$, we define the ith *partition rank* $r_i(G)$ of a given matrix G (with entries from some field) as

$$r_i(G) = r(A_i) + r(B_{n-i}) \tag{4}$$

where G is partitioned as in (2). We also define $r_0(G) = r_n(G) = r(G)$. Note that $r_i(G)$ is preserved under elementary row operations, but not necessarily under column permutations that interchange columns of A_i with columns of B_{n-i}. Define $r_{\max}(G) = \max_i r_i(G)$ to be the *maximum partition rank*.

The problem of finding a coordinate permutation to achieve $s(C)$ for a given code C is equivalent to finding a permutation of the columns of a generator matrix (or parity check matrix) for C with minimal maximum partition rank. For binary codes the decision problem corresponding to this optimization problem may be posed as follows:

Problem: PARTITION RANK PERMUTATION
Instance: A binary $k \times n$ matrix M, an integer s.
Question: Can M be transformed by column permutations into a matrix M' having maximum partition rank $r(M') \leq s$?

We also define some related problems:

Problem: iTH PARTITION RANK PERMUTATION
Instance: A binary $k \times n$ matrix M, an integer s, an integer i.
Question: Can M be transformed by column permutations into a matrix M' having ith partition rank $r_i(M') \leq s$?

Problem: PARTITION RANK VECTOR
Instance: A binary $k \times n$ matrix M, a vector of integers $(s_0, s_1, \ldots, s_n)$.
Question: Can M be transformed by column permutations into a matrix M' have ith partition rank $r_i(M') \leq s_i$ for all i, $0 \leq i \leq n$?

It is easy to see that all of these problems are in NP, as the partition rank vector of any candidate solution is easily obtained in polynomial time, using standard algorithms from linear algebra. In the case of PARTITION RANK PERMUTATION and iTH PARTITION RANK PERMUTATION, if there were a polytime algorithm for the decision problem, there would also be a polytime algorithm for the optimization problem: one would simply input $s = 1, 2, 3, \ldots$ to the decision problem, and retain the solution corresponding to the smallest s for which the decision algorithm returns an affirmative answer.

We have the following theorem.

Theorem 1 *iTH PARTITION RANK PERMUTATION is NP-complete.*

Proof: Restrict the input to be the incidence matrix of a graph and then reduce from SIMPLE MAX CUT [13], as sketched in Appendix A. □.

Note the NP-completeness of iTH PARTITION RANK PERMUTATION implies that PARTITION RANK VECTOR is also NP-complete. An input to the former problem is easily transformed to an input to the latter problem by taking $s_i = s$ and $s_j = \infty, j \neq i$. Unfortunately, neither result establishes the NP-completeness of PARTITION RANK PERMUTATION.

4 The Heuristic Search Procedure

In this section, we describe a simple heuristic descent procedure that can be used to find a "good" coordinate permutation π so that $s_{\max}(\pi C)$ is made small. Let C be a given binary code of length n. Let π be a candidate coordinate permutation, and $(s_0(\pi C), \ldots, s_n(\pi C))$ be the corresponding state complexity profile.

We define two objective functions: a primary objective and a secondary objective. In the process of our search, we accept any candidate solution which improves upon the best-seen primary objective value, or which matches the best-seen primary objective value,

and improves upon the corresponding secondary objective value. These two objective functions are, respectively,

$$\begin{aligned} f_1(\pi C) &= s_{\max}(\pi C) = \max_i \{s_i(\pi C)\}, \text{ and} \\ f_2(\pi C) &= \sum_i s_i(\pi C), \end{aligned}$$

i.e., the state complexity and the sum of the components of the state dimension profile, respectively.

A general "greedy" descent procedure operates as follows.

1. Choose a starting point in the search space, evaluate the objective function for this point, and record this point as the "best-seen" solution. Let the "current point" be this starting point.

2. Evaluate the objective function (or a related heuristic) for a set of candidates located in a neighborhood of the current point.

3. If no improvement is found, output the current point as a tentative "local optimum," otherwise, choose the best of the candidates as the new current point, update the record of best-seen solution, and return to step 2.

The greedy descent can be started from randomly chosen starting points and the best overall solution chosen.

In our case, the search space is the set of all possible coordinate permutations, which corresponds to the full symmetric group on n elements, where n is the code block length. Given a generator (or parity check) matrix for a code, we take as our "neighborhood" the set of all matrices that can be reached from the given matrix by transposing any two columns.

Unfortunately, direct evaluation of the objective function for each point in a given neighborhood, although polynomial in complexity, is still quite costly. To guide our search for local optima, we use the following heuristic objective function: we choose the column transposition that reduces the span length of the atomic generators in the trellis-oriented generator matrix as much as possible.

The terminology used in the previous sentence is defined as follows. In [3], but without giving a precise definition, Forney pointed out the significance of a particular form—the so-called trellis-oriented form—for the generator matrix of a linear code. For a given nonzero n-tuple $g = (g_1, \ldots, g_n)$, define the $start(g)$ as the smallest i for which $g_i \neq 0$ and the $end(g)$ as the largest i for which $g_i \neq 0$. Then g is said to have *span* $[start(g), end(g)]$, and span length $end(g) - start(g) + 1$. In [11] it is shown that the rows of the trellis-oriented generator matrix should be composed of atomic generators, i.e., codewords that cannot be expressed as linear combinations of codewords of strictly smaller span length. The minimal trellis can then be constructed by taking the product of elementary trellises corresponding to the one-dimensional subcodes generated by each row of the trellis-oriented generator matrix. A necessary and sufficient condition for a given generator matrix to be in trellis-oriented form is that no two rows of the matrix should either start in the same position or end in the same position [11].

A generator with span $[a, b]$ is said to be *active* at all times i, $a < i < b$. A generator that is active at time i contributes a dimension to the state space at that time; consequently the state space dimension s_i at time i is equal to the number of atomic

generators active at time i. The intuition in attempting to reduce the span lengths of the atomic generators is to reduce the amount of time the generator is active, and hence the number of dimensions that the generator contributes.

If g_i denotes the ith row of the trellis-oriented generator matrix, then it is easy to see for the code πC that

$$\sum_{i=1}^{k}[end(g_i) - start(g_i)] = \sum_{i=0}^{n} s_i(\pi C) = f_2(\pi C) \tag{5}$$

Thus our heuristic is, in fact, attempting to achieve the secondary objective.

Our algorithm proceeds by evaluating the effect of each column transposition on the span length sum of the atomic generators and chooses the transposition that reduces this sum as much as possible. A "heuristic local optimum" is reached when no column transpositions cause a reduction in the span length sum.

When a heuristic local optimum is reached, actual evaluation of the objective functions may show that further reduction in either or both functions may be possible. We perform such exploration whenever the heuristic local optimum is sufficiently "good." In this way, excessive (and costly) evaluation of the true objective functions at numerous points in the course of descent is avoided.

5 Results

A small table of results, listing various codes, their usual parameters, the state complexity for the best code permutation found and the corresponding state complexity profile is given on the next page. State complexity profiles that match the DLP lower bound [7] are marked with an asterisk; these profiles are componentwise optimal, i.e., in no position can they be reduced. Other profiles are marked with a caret symbol '^ ' in those positions for which the corresponding state complexity exceeds the DLP lower bound, which, however, need not be achievable [7]. Suboptimal codes, for which the DLP lower bound does not apply directly, are marked with a †.

For the extended BCH codes are results on $s_{\max}$ in all cases match (without improving) the results of Vardy and Be'ery [6]. It is interesting to note that the lexicodes have better state complexity profiles than the BCH codes with the same parameters. In fact, the (16,7,6) lexicode has the optimum state complexity profile.

6 Conclusions

In this paper we have presented a heuristic search procedure that may be used to search for "good" coordinate permutations of binary linear codes, with the aim of reducing the state complexity. We have applied this procedure to a small set of codes with encouraging results. Work is underway to identify classes of codes which achieve a small trellis complexity for given n, k, and d.

An initial goal of this paper was to prove the conjecture that the MAXIMUM PARTITION RANK problem is NP-complete. So far we have only been able to show that the related iTH PARTITION RANK problem is NP-complete, and so propose establishing the computational complexity class of the former problem as a topic for further research.

Code	(n,k,d)	$s_{\max}$	State Complexity Profile
BCH	(15,5,7)	4	0-1-2-3-3-4-4-4-$\hat{4}$-3-4-4-3-3-2-1-0
BCH	(15,7,5)	5	0-1-2-3-3-4-$\hat{4}$-$\hat{5}$-$\hat{5}$-$\hat{5}$-$\hat{5}$-$\hat{4}$-3-3-2-1-0
lexicode	(15,7,5)	4	0-1-2-3-3-$\hat{4}$-4-4-4-4-$\hat{4}$-3-3-2-1-0
BCH	(15,11,3)	4	0-1-2-2-3-3-$\hat{4}$-$\hat{4}$-$\hat{4}$-3-3-3-2-2-1-0
ext BCH	(16,5,8)	4	0-1-2-3-3-4-4-4-3-4-4-4-3-3-2-1-0*
ext BCH	(16,7,6)	6	0-1-2-3-3-4-$\hat{5}$-$\hat{6}$-5-$\hat{6}$-$\hat{5}$-4-3-3-2-1-0
lexicode	(16,7,6)	5	0-1-2-3-3-4-4-4-5-4-4-4-3-3-2-1-0*
ext BCH	(16,11,4)	4	0-1-2-3-3-4-4-4-3-4-4-4-3-3-2-1-0*
QR	(17,9,5)	5	0-1-2-3-4-4-$\hat{5}$-5-5-5-5-5-$\hat{5}$-4-4-3-2-1-0
ext QR	(18,9,6)	6	0-1-2-3-4-5-$\hat{5}$-$\hat{6}$-6-6-$\hat{6}$-6-$\hat{6}$-$\hat{5}$-5-4-3-2-1-0
BCH	(31,6,15)	5	0-1-2-3-3-4-4-4-4-5-5-5-5-5-5-5-$\hat{5}$-4-5-5-5-5-5-5-4-4-4-4-3-3-2-1-0
BCH	(31,11,11)	10	0-1-2-3-4-5-5-6-$\hat{7}$-7-8-$\hat{8}$-$\hat{9}$-$\hat{9}$-$\hat{10}$-$\hat{10}$-$\hat{10}$-$\hat{9}$-$\hat{9}$-$\hat{9}$-$\hat{8}$-8-7-$\hat{7}$-6-5-5-4-3-2-
BCH	(31,16,7)	9	0-1-2-3-4-5-6-6-7-8-9-8-9-8-7-6-7-8-9-8-9-8-7-6-7-6-5-4-3-2-1-0†
BCH	(31,21,5)	10	0-1-2-3-4-4-5-6-$\hat{6}$-$\hat{7}$-$\hat{7}$-$\hat{8}$-$\hat{9}$-$\hat{10}$-$\hat{9}$-$\hat{9}$-$\hat{9}$-$\hat{8}$-$\hat{8}$-$\hat{8}$-$\hat{7}$-$\hat{7}$-$\hat{6}$-6-5-5-4-3-2-1-
BCH	(31,26,3)	5	0-1-2-2-3-3-3-3-4-4-4-4-4-4-$\hat{5}$-5-$\hat{5}$-5-$\hat{5}$-5-$\hat{5}$-4-4-4-4-4-3-3-3-2-2-1-0
ext BCH	(32,6,16)	5	0-1-2-3-3-4-4-4-4-5-5-5-5-5-5-5-4-5-5-5-5-5-5-5-4-4-4-4-3-3-2-*
ext BCH	(32,11,12)	10	0-1-2-3-4-5-5-6-$\hat{7}$-$\hat{8}$-$\hat{9}$-$\hat{10}$-$\hat{9}$-$\hat{10}$-$\hat{10}$-$\hat{10}$-$\hat{10}$-$\hat{10}$-$\hat{10}$-$\hat{9}$-$\hat{9}$-$\hat{9}$-8-8-7-$\hat{7}$-6-5-$\hat{5}$-
ext BCH	(32,16,8)	9	0-1-2-3-4-5-6-7-6-7-8-9-8-9-8-7-6-7-8-9-8-9-8-9-8-7-6-7-6-5-4-3-2-†
ext BCH	(32,21,6)	10	0-1-2-3-4-5-5-6-$\hat{7}$-7-8-$\hat{9}$-$\hat{9}$-$\hat{10}$-$\hat{10}$-$\hat{10}$-9-$\hat{10}$-$\hat{10}$-$\hat{10}$-$\hat{9}$-$\hat{9}$-8-7-$\hat{7}$-6-5-$\hat{5}$-
ext BCH	(32,26,4)	5	0-1-2-3-3-4-4-4-4-5-5-5-5-5-5-5-4-5-5-5-5-5-5-5-4-4-4-4-3-3-2-*

A An NP-Complete Problem

In this appendix we sketch a proof that iTH PARTITION RANK PERMUTATION is NP-complete, by reducing from the SIMPLE MAX CUT [13] problem (see below). Space constraints force us to omit some details, which may be found in [14]. We use the same notation as in Section 3. By restricting the input to the iTH PARTITION RANK PERMUTATION problem to be the incidence matrix of a graph, we obtain a graph-theoretic version of the problem.

Specifically, let $G = (V, E)$ be a graph with n edges, m vertices and no self-loops. Order the edges as $\{e_1, \ldots, e_n\}$ and the vertices as $\{v_1, \ldots, v_m\}$. Define an $m \times n$ incidence matrix $M = [m_{ij}]$ whose m rows correspond to the m vertices and whose n columns correspond to the n edges, as follows:

$$m_{ij} = \begin{cases} 1 & \text{if the } e_j \text{ is incident on } v_i, \textit{and} \\ 0 & \text{otherwise} \end{cases}$$

The rank $r(G)$ of a graph G is defined as the number of vertices m minus the number of connected components of G. If M is an incidence matrix for G, then [15]

$$r(G) = r(M).$$

Let $E_i = \{e_1, \ldots, e_i\}$ be the set containing the "first" i edges of E, and let E_i induce two subgraphs $G_1(i) = (V_1, E_i)$ and $G_2(i) = (V_2, E - E_i)$. Define the ith "graph partition rank" as

$$r_i(G) = r(G_1(i)) + r(G_2(i)).$$

It is now straightforward to translate questions about the graph partition rank into questions about the matrix partition rank of the corresponding incidence matrix, and vice-versa.

We begin by making some relevant problem definitions.

Problem: SIMPLE MAX CUT
Instance: A graph $G = (V, E)$, a positive integer w.
Question: Is there a set $S \subset V$ such that $|\{\{u, v\} \in E : u \in S, v \in V - S\}| \geq w$?
Remark: It is well known [8, 13] that SIMPLE MAX CUT is NP-complete.

Problem: GRAPH PARTITION INTO EQUAL-SIZED SUBGRAPHS OF MINIMUM RANK
Instance: A graph $G = (V, E)$, a positive integer w.
Question: Can E be partitioned into two disjoint sets E_1, $E - E_1$, with $|E_1| = |E - E_1|$, so that the disjoint induced subgraphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ have rank sum satisfying

$$r(G_1) + r(G_2) \leq w?$$

Remark: Evidently we are asking if the edges of G can be arranged in some order so that $r_{n/2}(G) \leq w$. We refer to a partition of the edges of a graph as in this problem as an *edge partition.*

Before proving that this problem is NP-complete, we make the following observations.

Suppose an edge partition of a graph $G = (V, E)$ induces a pair of (not necessarily connected) subgraphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ have I_1 and I_2 connected components, respectively. Then, since $r(G_1) = |V_1| - I_1$, $r(G_2) = |V_2| - I_2$ and $|V_1| + |V_2| = |V_1 \cup V_2| + |V_1 \cap V_2| = |V| + |V_1 \cap V_2|$ it follows that the rank sum can be expressed as

$$r(G_1) + r(G_2) = |V| + |V_1 \cap V_2| - (I_1 + I_2). \tag{6}$$

If G_1 and G_2 are both constrained to be connected, then $I_1 = I_2 = 1$, and minimizing the rank sum is equivalent to minimizing $|V_1 \cap V_2|$, the number of vertices the two subgraphs have in common.

We now prove the main result.

Theorem 2 GRAPH PARTITION INTO EQUAL-SIZED SUBGRAPHS OF MINIMUM RANK *is NP-complete.*

Sketch of Proof:[1] It is easy to see that this problem is in NP since a nondeterministic algorithm need only guess a subset E_1 of appropriate size from E and check in polynomial time whether $r(G_1) + r(G_2) \leq w$. We now show that there is a polynomial transformation from SIMPLE MAX CUT to GRAPH PARTITION INTO EQUAL-SIZED SUBGRAPHS OF MINIMUM RANK. For convenience, we refer to the former problem as SMC and the latter problem as GPE.

Given a graph $G = (V, E)$ and a positive integer w as input for SMC,

1. let $m = |V|$, $n = |E|$, and $U = \{u_1, \ldots, u_m\}$ satisfy $U \cap V = \emptyset$;
2. let $S = V \cup U$ and let $T = \{\{s_i, s_j\} : s_i, s_j \in S \text{ and } s_i, s_j \notin E\}$ be the complement of E in the edge set of the complete graph on S;
3. relabel the vertices of S as $s_1, \ldots, s_{2m}$, where s_i corresponds to v_i, $1 \leq i \leq m$, and s_{m+i} corresponds to u_i, $1 \leq i \leq m$;
4. then, corresponding to each vertex in $s_i \in S$, form a complete graph $S_i = (C_i, A_i)$ on m^3 vertices, where $C_i = \{c_{i,1}, \ldots, c_{i,m^3}\}$, and $A_i = \{\{a, b\} : a, b \in C_i\}$;
5. finally, if $n - m$ is odd let $X = \{x_1, x_2\}$ be a set of two vertices and $Y = \{X\}$ an edge connecting them, otherwise let $X = \emptyset$ and $Y = \emptyset$.

[1] We are constrained by space limitations from giving the complete details, which are, however available in [14].

The corresponding input for GPE is the graph $G' = (V', E')$ and positive integer w' defined as

$$\begin{aligned} V' &= C_1 \cup \cdots \cup C_{2m} \cup X; \\ E' &= A_1 \cup \cdots \cup A_{2m} \cup Y \cup \{\{c_{i,j}, c_{j,i}\} : \{v_i, v_j\} \in T\}; \\ w' &= 2m^4 + m^2 - w + |Y| - 2. \end{aligned}$$

This construction is an extended version of the one used by Garey, Johnson and Stockmeyer [13] in their transformation of SMC to MINIMUM CUT INTO EQUAL-SIZED SUBSETS. In that problem, the *vertices* of a graph must be partitioned into two disjoint sets of equal size with minimum possible number of *edges* in common. In light of the remark following (6), we must partition the *edges* of G' into disjoint sets of equal size with the minimum possible number of *vertices* in common.

The main addition to the construction of [13] is the substitution of the original $2m$ vertices created by their transformation with $2m$ complete graphs on m^3 vertices, $S_1, \ldots, S_{2m}$. The edge Y in the construction merely ensures that G' has an even number of edges. The construction can obviously be performed in polytime because there are $O(m^4)$ vertices in the final resulting graph.

Suppose now that the original SMC problem has a solution, i.e., that there is a partition $V = V_1 \cup V_2$ such that $|\{\{u, v\} \in E : u \in V_1, v \in V_2\}| \geq w$. We must show that E' can be split into two disjoint subsets of equal size to induce two subgraphs with rank sum $\leq w'$. First, we partition the vertices of the S_i to form two subsets N_1 and N_2, as follows. Let

$$N_1 = \Big(\bigcup_{i:v_i \in V_1} C_i\Big) \cup \Big(\bigcup_{i:v_i \in V_2} C_{m+i}\Big); \quad N_2 = \Big(\bigcup_{i:v_i \in V_2} C_i\Big) \cup \Big(\bigcup_{i:v_i \in V_1} C_{m+i}\Big)$$

and let G_1 and G_2 be the corresponding induced subgraphs. Note that G_1 and G_2 are connected, and each has rank $r(G_1) = r(G_2) = m^4 - 1$. Not counting Y, there remain

$$\begin{aligned} t &= |\{\{u, v\} \in E' : u \in N_1, v \in N_2\}| \\ &= m^2 - |\{\{u, v\} \notin E' : u \in N_1, v \in N_2\}| \\ &= m^2 - |\{\{u, v\} \in E : u \in V_1, v \in V_2\}| \\ &\leq m^2 - w \end{aligned}$$

unassigned edges. If the difference between the number of edges in G_1 and G_2 is smaller than or equal to $t + |Y|$, these remaining edges can be assigned (arbitrarily) to G_1 and G_2 to form G_1' and G_2', each having an equal number of edges with any imbalance made up by appropriate assignment of Y. Since none of these edges have any vertices in common, this adds approximately $t/2$ vertices to each of G_1 and G_2; in fact, it is easy to see that

$$r(G_1') + r(G_2') = 2m^4 - 2 - |Y| \leq 2m^4 + m^2 - w + |Y| - 2 = w'.$$

The argument showing that the difference between the number of edges in G_1 and G_2 is $\leq t + |Y|$ is given in [14].

Conversely, suppose now that there is a solution to the GPE problem, i.e., a partition of G' into two disjoint subgraphs $G_1' = (V_1', E_1')$ and $G_2' = (V_2', E_2')$ with $E' = E_1' \cup E_2'$, $E_1' \cap E_2' = \emptyset$ and $|E_1'| = |E_2'| = i$, such that the ith graph partition rank $r_i(G') = r(G_1') + r(G_2') \leq w' = 2m^4 + m^2 - w + |Y| - 2$. Remove the isolated edge Y (if it exists) from G' as it is no longer relevant. There are $2m^4$ vertices in G', so, from (6), it follows that $|V_1' \cap V_2'| \leq m^2 - w - 2 + I_1 + I_2$, where I_1 and I_2 are the number of components of G_1 and G_2, respectively. In [14], we show that (for $m > 8$) no optimum GPE solution would "split" any of the complete subgraphs S_i, because of the large rank this would cause the G_1' and G_2' to have. Furthermore, we show that any solution can be re-arranged (without sacrificing optimality) into one in which G_1' and G_2' are each connected, i.e., $I_1 = I_2 = 1$. Since then the two subgraphs have $|V_1' \cap V_2'| \leq m^2 - w$ vertices in common, the "inverse transformation" shows that G must have a cut of size w or greater.

References

[1] J. K. Wolf, "Efficient maximum-likelihood decoding of linear block codes using a trellis," *IEEE Trans. on Inform. Theory*, vol. IT-24, pp. 76–80, 1978.

[2] J. L. Massey, "Foundation and methods of channel encoding," in *Proc. Int. Conf. Inform. Theory and Systems*, vol. 65, (Berlin), Sept. 1978.

[3] G. D. Forney, Jr., "Coset codes II: Binary lattices and related codes," *IEEE Trans. on Inform. Theory*, vol. 34, pp. 1152–1187, Sep. 1988.

[4] T. Kasami, T. Takata, T. Fujiwara, and S. Lin, "On the optimum bit orders with respect to the state complexity of trellis diagrams for binary linear codes," *IEEE Trans. on Inform. Theory*, vol. 39, pp. 242–245, Jan. 1993.

[5] T. Kasami, T. Takata, T. Fujiwara, and S. Lin, "On complexity of trellis structure of linear block codes," *IEEE Trans. on Inform. Theory*, vol. 39, pp. 1057–1064, May 1993.

[6] A. Vardy and Y. Be'ery, "Maximum-likelihood soft decision decoding of BCH codes," *IEEE Trans. Inform. Theory*, vol. 40, pp. 546–554, March 1994.

[7] G. D. Forney, Jr., "Dimension/length profiles and trellis complexity of linear block codes." Submitted to *IEEE Trans. Inform. Theory*, October 1993.

[8] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness.* New York: W. H. Freeman and Company, 1979.

[9] J. H. Conway and N. J. A. Sloane, "Lexicographic codes: Error-correcting codes from Game Theory," *IEEE Trans. on Inform. Theory*, vol. IT-32, pp. 337–348, May 1986.

[10] D. J. Muder, "Minimal trellises for block codes," *IEEE Trans. on Inform. Theory*, vol. 34, pp. 1049–1053, Sept. 1988.

[11] F. R. Kschischang and V. Sorokine, "On the trellis structure of block codes." Submitted to *IEEE Trans. on Inform Theory*, Aug. 1993.

[12] G. D. Forney, Jr. and M. D. Trott, "The dynamics of group codes: State spaces, trellis diagrams and canonical encoders," *IEEE Trans. on Inform. Theory*, vol. 39, pp. 1491–1513, Sept. 1993.

[13] M. R. Garey, D. S. Johnson, and L. Stockmeyer, "Some simplified NP-complete graph problems," *Theoretical Computer Science*, vol. 1, pp. 237–267, 1976.

[14] G. B. Horn and F. R. Kschischang, "The NP-completeness of the ith partition rank permutation problem," Tech. Rep. 94-2, University of Toronto, Dept. of Electrical and Computer Engineering, September 1994.

[15] N. Deo, *Graph Theory with Applications to Engineering and Computer Science.* Englewood Cliffs, N.J.: Prentice-Hall, 1974.

Error-and-Erasure Decoding of Golay Codes via Code Decomposition *

B.-Z. Shen and K.K. Tzeng
Department of Electrical Engineering and Computer Science
Lehigh University, Bethlehem, PA 18015-3084

Abstract

Algorithms for error-and-erasure decoding of binary Golay codes up to their actual minimum distance are developed through a code decomposition approach. Thus, generalized minimum distance decoding of the $(24, 12, 8)$ Golay code up to actual minimum distance 8 becomes feasible. This in turn should offer a more efficient algorithm for bounded-distance decoding of Leech lattice.

I Introduction

It is well known that soft decision decoding of binary Golay codes plays a crucial part in block coded modulation. During the last decade, many soft decision decoding algorithms of the binary Golay codes have been developed. The main part of almost all these algorithms is trellis decoding, that is searching. In [1], Forney mentioned that by using a generalized minimum distance (GMD) decoder, which uses several trials of error-and-erasure decoding, to decode a Golay code, one may reduce the complexity without much loss on coding gain compared to those non-optimal fast soft decoding algorithms. However, it appears that all the known hard decision decoding algorithms of a Golay code cannot be applied toward error-and-erasure decoding up to their actual minimum distance.

In this paper, through a code decomposition approach, we present error-and-erasure decoding algorithms for the binary (23,12,7) and (24,12,8) Golay codes, denoted by $\mathcal{G}_{23}$ and $\mathcal{G}_{24}$ respectively, up to their actual minimum distance.

To decode $\mathcal{G}_{23}$, we decompose this code into the direct sum of a repetition code and another subcode. Associated with this decomposition, a partial transform is defined. It is shown that, using the partial transform, the decoding of $\mathcal{G}_{23}$ can eventually be reduced to a trivial decoding of the repetition code.

For any codeword in $\mathcal{G}_{24}$, it is known that, by deleting any of its components, the result is a codeword in $\mathcal{G}_{23}$. Using this property and the decoding algorithm for $\mathcal{G}_{23}$, an error-and-erasure decoding algorithm for $\mathcal{G}_{24}$ is also derived.

*This work was supported by the National Science Foundation under Grants NCR-9016095 and NCR-9406043.

II Error-and-erasure decoding of $\mathcal{G}_{23}$

A. Binary Golay code $\mathcal{G}_{23}$ and its decomposition

Let α be a primitive 23rd root of unity in $\mathbf{F}_{2^{11}}$. Denote $m_i(x)$ the minimal polynomial of α^i. Then $\mathcal{G}_{23}$ is a $(23, 12, 7)$ binary cyclic code with generator polynomial

$$g(x) = m_1(x) = x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1.$$

Its complete defining set is

$$\{\alpha^1, \alpha^2, \alpha^3, \alpha^4, \alpha^6, \alpha^8, \alpha^9, \alpha^{12}, \alpha^{13}, \alpha^{16}, \alpha^{18}\}.$$

For convenience, we also refer to $R := \{1, 2, 3, 4, 6, 8, 9, 12, 13, 16, 18\}$ as the complete defining set of $\mathcal{G}_{23}$. Let

$$\tilde{R} := \{5 \cdot 2^k \bmod 23 | k \geq 0\} = \{5, 7, 10, 11, 14, 15, 17, 19, 20, 21, 22\}.$$

Let C_1 be the binary cyclic code of length 23 generated by

$$m_1(x)m_5(x) = (x^{23} - 1)/(x - 1).$$

Then C_1 is a subcode of $\mathcal{G}_{23}$ and

$$C_1 = \{(00\cdots0), (11\cdots1)\},$$

namely a $(23, 1, 23)$ repetition code. To construct another subcode of $\mathcal{G}_{23}$, we define

$$u_i(x) := x^{i-1}g(x), i = 1, \ldots, 11.$$

Clearly, $u_i(x) \in \mathcal{G}_{23} \setminus C_1$ and $u_1(x), \ldots, u_{11}(x)$ are linearly independent. Furthermore, it can be easily verified that, for $i \in \tilde{R}$, the following matrix

$$U(\alpha^i) = \begin{bmatrix} u_1(\alpha^i) & u_1(\alpha^{2i}) & \cdots & u_1(\alpha^{i2^{10}}) \\ u_2(\alpha^i) & u_2(\alpha^{2i}) & \cdots & u_2(\alpha^{i2^{10}}) \\ \vdots & & \ddots & \vdots \\ u_{11}(\alpha^i) & u_{11}(\alpha^{2i}) & \cdots & u_{11}(\alpha^{i2^{10}}) \end{bmatrix}$$

is nonsingular [2]. Let $\boldsymbol{u}_i = (u_{i0}, \ldots, u_{22})$ be the vector corresponding to $u_i(x)$. Then we define

$$C_2 := \langle \boldsymbol{u}_1, \ldots, \boldsymbol{u}_{11} \rangle,$$

a subcode of $\mathcal{G}_{23}$ generated by $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{11}$.

We now have the following decomposition

$$\mathcal{G}_{23} = C_1 \oplus C_2$$

that is for every $\boldsymbol{c} \in \mathcal{G}_{23}$, there exist unique $\boldsymbol{c}_i \in C_i$ for $i = 1, 2$, such that $\boldsymbol{c} = \boldsymbol{c}_1 + \boldsymbol{c}_2$.

Let $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e} + \boldsymbol{f}$ be a received vector, where $\boldsymbol{c} \in \mathcal{G}_{23}$, $\boldsymbol{e}$ is the error vector and $\boldsymbol{f}$ is the erasure vector. Then,

$$\boldsymbol{r} = \boldsymbol{c}_1 + \boldsymbol{c}_2 + \boldsymbol{e} + \boldsymbol{f}.$$

The decoding procedure is then divided into the following two parts.

Part 1. Find $\boldsymbol{c}_2$ from $\boldsymbol{r}$;

Part 2. Find $\boldsymbol{c}_1$ from $\boldsymbol{r}_1 := \boldsymbol{r} - \boldsymbol{c}_2$, and compute $\boldsymbol{c} = \boldsymbol{c}_1 + \boldsymbol{c}_2$.

It is easy to see that Part 2 is trivial since C_1 is a repetition code. To accomplish Part 1, we will use a spectral technique introduced in the next subsection.

B. Partial Transform

For every $\boldsymbol{b} \in \mathbf{F}_2^{23}$, denote $b(x) = \sum_{i=0}^{22} b_i x^i$. Then for any $i \in \{0, 1, \ldots, 22\}$, define a partial transform PT_i from $\mathbf{F}_2^{23}$ to $\mathbf{F}_{2^{11}}^{11}$ as follows

$$PT_i(\boldsymbol{b}) := (b(\alpha^i), b(\alpha^{2i}), \ldots, b(\alpha^{2^{10}i})).$$

It is known that the Fourier transform of $\boldsymbol{b}$ is

$$(b(\alpha^0), b(\alpha^1) \ldots, b(\alpha^{22})).$$

Thus, $PT_i(\boldsymbol{b})$ is in fact a part of the Fourier transform.

It is also known that one of the advantages of using Fourier transform is that it has an inverse transform. Although the partial transform defined has no inverse on $\mathbf{F}_2^{23}$, it does have an inverse on C_2, when $i \in \tilde{R}$.

Let $\boldsymbol{c}_2 \in C_2$ and $i \in \tilde{R}$, then there exist $m_1, \ldots, m_{11}$ such that

$$\boldsymbol{c}_2 = (m_1, \ldots, m_{11})\mathbf{U}, \quad \text{where } \mathbf{U} = \begin{pmatrix} \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_{11} \end{pmatrix}.$$

Then,

$$PT_i(\boldsymbol{c}_2) = (m_1, \ldots, m_{11})U(\alpha^i).$$

Thus

$$PT_i(\boldsymbol{c}_2)U(\alpha^i)^{-1}\mathbf{U} = (m_1, \ldots, m_{11})\mathbf{U} = \boldsymbol{c}_2.$$

Therefore, we can define an inverse map PT_i^{-1} as follows

$$PT_i^{-1}(\boldsymbol{y}) := \boldsymbol{y}U(\alpha^i)^{-1}\mathbf{U}, \text{ for } \boldsymbol{y} \in \mathbf{F}_{2^{11}}^{23}.$$

Theorem 1 *For $i \in \tilde{R}$,*

$$PT_i^{-1}(PT_i(\boldsymbol{u})) = \boldsymbol{u} \text{ for } \boldsymbol{u} \in C_2.$$

As it is stated in Subsection A, the crucial part of our decoding approach is to find $\boldsymbol{c}_2$. By the above theorem, it is enough to find $PT_i(\boldsymbol{c}_2)$ for an $i \in \tilde{R}$. Since we know only the received vector $\boldsymbol{r}$ and the erasure locations, let us first look at the relation between $PT_i(\boldsymbol{r})$ and $PT_i(\boldsymbol{c}_2)$. Let $r(x), c_1(x), c_2(x), e(x)$ and $f(x)$ be polynomials corresponding to the vectors $\boldsymbol{r}, \boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{e}$ and $\boldsymbol{f}$, respectively. Then we have

$$PT_i(\boldsymbol{r}) = (r(\alpha^i), r(\alpha^i)^2, \ldots, r(\alpha^i)^{2^{10}})$$

$$PT_i(\boldsymbol{c}_2) = (c_2(\alpha^i), c_2(\alpha^i)^2, \ldots, c_2(\alpha^i)^{2^{10}}),$$

since $\boldsymbol{r}, \boldsymbol{c}_2 \in \mathbf{F}_2^{23}$. Thus, it is enough to find the relation between $r(\alpha^i)$ and $c_2(\alpha^i)$ which can be stated as follows

$$r(\alpha^i) = c_2(\alpha^i) + e(\alpha^i) + f(\alpha^i) \tag{1}$$

since $\boldsymbol{r} = \boldsymbol{c}_1 + \boldsymbol{c}_2 + \boldsymbol{e} + \boldsymbol{f}$ and the defining set of C_1 contains $\tilde{R}$. Clearly, for every $j \in R$ we have $c_2(\alpha^j) = c_1(\alpha^j) = 0$. Thus, we have

$$r(\alpha^j) = e(\alpha^j) + f(\alpha^j), \text{ for } j \in R.$$

Therefore, if we can find a linear combination of $f(\alpha^j)$ for $j \in R$, say $\sum_{j\in R} \beta_j f(\alpha^j)$ such that

$$f(\alpha^i) + \sum_{j\in R} \beta_j f(\alpha^j) = 0 \tag{2}$$

then

$$\begin{aligned} c_2(\alpha^i) &= [r(\alpha^i) + \sum_{j\in R} \beta_j r(\alpha^j)] + [e(\alpha^i) + \sum_{j\in R} \beta_j e(\alpha^j)] + [f(\alpha^i) + \sum_{j\in R} \beta_j f(\alpha^j)] \\ &= [r(\alpha^i) + \sum_{j\in R} \beta_j r(\alpha^j)] + [e(\alpha^i) + \sum_{j\in R} \beta_j e(\alpha^j)]. \end{aligned}$$

Hereafter, for any two vectors $\boldsymbol{x} = (x_0, \ldots, x_{22})$ and $\boldsymbol{y} = (y_0, \ldots, y_{22}) \in \mathbf{F}_{2^{11}}^{23}$, denote

$$\boldsymbol{x} \cdot \boldsymbol{y} := \sum_{i=0}^{22} x_i y_i \text{ and } \boldsymbol{x} * \boldsymbol{y} := (x_0 y_0, \ldots, x_{22} y_{22}).$$

Furthermore, define

$$v(\alpha^k) := (1, \alpha^k, (\alpha^k)^2, \ldots, (\alpha^k)^{22}) \text{ for } k \in \{0, 1, \ldots, 22\}$$

Then, we have

$$\boldsymbol{f} \cdot [v(\alpha^i) + \sum_{j\in R} \beta_j v(\alpha^j)] = f(\alpha^i) + \sum_{j\in R} \beta_j f(\alpha^j).$$

Therefore, what we need to do is to find $\beta_j \in \mathbf{F}_{2^{11}}$ with $j \in R$ such that

$$\boldsymbol{f} \cdot [v(\alpha^i) + \sum_{j\in R} \beta_j v(\alpha^j)] = 0.$$

Since the only possible nonzero components of $\boldsymbol{f}$ are at the erasure locations, it is sufficient to find linear combination of vectors $v(\alpha^j)$ for $j \in R$, such that the components of $v(\alpha^i) + \sum_{j\in R} \beta_j v(\alpha^j)$ at erasure locations are zero. To this end, let us define the following two vector sequences

$$\boldsymbol{a}_1 = v(1), \boldsymbol{a}_2 = v(\alpha^1), \boldsymbol{a}_3 = v(\alpha^4), \boldsymbol{a}_4 = v(\alpha^{16}), \boldsymbol{a}_5 = v(\alpha^2), \boldsymbol{a}_6 = v(\alpha^3), \boldsymbol{a}_7 = v(\alpha^5);$$

$$\boldsymbol{b}_1 = v(1), \boldsymbol{b}_2 = v(\alpha^2), \boldsymbol{b}_3 = v(\alpha^8), \boldsymbol{b}_4 = v(\alpha^{12}), \boldsymbol{b}_5 = v(\alpha), \boldsymbol{b}_6 = v(\alpha^4), \boldsymbol{b}_7 = v(\alpha^5).$$

These two sequences have the following property:

a) $\boldsymbol{a}_u * \boldsymbol{b}_v = v(\alpha^j)$ for a $j \in R \cup \{0\}$), if $u + v \leq 7$;

b) $\boldsymbol{a}_u * \boldsymbol{b}_v = v(\alpha^i)$ for a $i \in \tilde{R}$, if $u + v = 8$.

For any given erasure locations $k_1, \ldots, k_\rho$ with $0 \leq k_1 < k_2 \cdots < k_\rho \leq 22$, where $1 \leq \rho \leq 6$. Define a projective map $P_{k_1,\ldots,k_\rho}$ from $\mathbf{F}_{2^{11}}^{23}$ to $\mathbf{F}_{2^{11}}^{\rho}$ as follows

$$P_{k_1,\ldots,k_\rho}(\boldsymbol{x}) = (x_{k_1}, \ldots, x_{k_\rho}), \quad \forall \ \boldsymbol{x} = (x_0, x_1, \ldots, x_{22}) \in \mathbf{F}_{2^{11}}^{23}.$$

Consider the vectors $P_{k_1,\ldots,k_\rho}(\boldsymbol{a}_1), \ldots, P_{k_1,\ldots,k_\rho}(\boldsymbol{a}_7) \in \mathbf{F}_{2^{11}}^{\rho}$. From the linear dependence relations among these 7 vectors, one can obtain $7-\rho$ vectors $\boldsymbol{a}_1^*, \ldots, \boldsymbol{a}_{7-\rho}^*$ defined by

$$\boldsymbol{a}_l^* := \boldsymbol{a}_{r_l} + \beta_{1,l}\boldsymbol{a}_{r_l-1} + \cdots + \beta_{r_l-1,l}\boldsymbol{a}_1, \ l = 1, \ldots, 7-\rho$$

such that

$$P_{k_1,\ldots,k_\rho}(\boldsymbol{a}_l^*) = \mathbf{0}, \ l = 1, \ldots, 7-\rho \tag{3}$$

Furthermore, one can obtain $\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_{7-\rho}^*$, a subsequence of $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_7$, such that

(i) $\boldsymbol{a}_u^* * \boldsymbol{b}_v^* = \sum_{j \in R \cup \{0\}} \beta_j v(\alpha^j)$, when $u + v \leq 7 - \rho$;

(ii) $\boldsymbol{a}_u^* * \boldsymbol{b}_v^* = v(\alpha^i) + \sum_{j \in R \cup \{0\}} \beta_j v(\alpha^j)$, when $u + v = 7 - \rho + 1$, where $i \in \tilde{R}$.

Moreover, from (3) we have

(iii) $P_{k_1,\ldots,k_\rho}(\boldsymbol{a}_u^* * \boldsymbol{b}_v^*) = \mathbf{0}$ for $u + v \leq 7 - \rho + 1$.

Theorem 2 *Let $i \in \tilde{R}$. Let $\boldsymbol{w}_i = (w_{i,0}, \ldots, w_{i,22}) = v(\alpha^i) + \sum_{j \in R \cup \{0\}} \beta_j(\alpha^j)$, and $w_{i,k_1} = \cdots = w_{i,k_\rho} = 0$. Then*

$$PT_i(\boldsymbol{c}_2) = (c_2(\alpha^i), c_2(\alpha^i)^2, \ldots, c_2(\alpha^i)^{2^{10}}).$$

with

$$c_2(\alpha^i) = \boldsymbol{r} \cdot \boldsymbol{w}_i + \boldsymbol{e} \cdot \boldsymbol{w}_i + \beta_0 S_0(\boldsymbol{c}),$$

and $S_0(\boldsymbol{c}) = \sum_{l=0}^{22} c_l$, which is 0 or 1.

Proof. Since $w_{i,k_1} = \ldots = a_{i,k_\rho} = 0$, we have $\boldsymbol{r} \cdot \boldsymbol{w}_i = (\boldsymbol{c} + \boldsymbol{e}) \cdot \boldsymbol{w}_i$. Thus

$$\boldsymbol{r} \cdot \boldsymbol{w}_i \ = \ (\boldsymbol{c} + \boldsymbol{e}) \cdot v(\alpha^i) + \boldsymbol{e} \cdot \Big(\sum_{j \in R \cup \{0\}} \beta_j v(\alpha^j) \Big) + \boldsymbol{c} \cdot \beta_0 v(1) \tag{4}$$

since $\boldsymbol{c} \cdot v(\alpha^j) = 0$ for $j \in R$. Besides, since $\boldsymbol{c} = \boldsymbol{c}_1 + \boldsymbol{c}_2$ and i is in the defining set of C_1, we have

$$(\boldsymbol{c} + \boldsymbol{e}) \cdot v(\alpha^i) = c_2(\alpha^i) + e(\alpha^i)$$

and $\boldsymbol{c} \cdot v(1) = S_0(\boldsymbol{c})$. Thus, (4) becomes

$$\boldsymbol{r} \cdot \boldsymbol{w}_i = c_2(\alpha^i) + \boldsymbol{e} \cdot \boldsymbol{w}_i + \beta_0 S_0(\boldsymbol{c}).$$

This implies that

$$c_2(\alpha^i) = \boldsymbol{r} \cdot \boldsymbol{w}_i + \boldsymbol{e} \cdot \boldsymbol{w}_i + \beta_0 S_0(\boldsymbol{c}).$$

□

According to the above theorem and the construction of $\boldsymbol{a}_k^*$ and $\boldsymbol{b}_k^*$, we have that, for any integer $l \leq 7-\rho$, if

$$\boldsymbol{a}_l^* * \boldsymbol{b}_{7-\rho-l+1}^* = v(\alpha^i) + \sum_{j \in R \cup \{0\}} \beta_{l,j} v(\alpha^j)$$

then $PT_i(\boldsymbol{c}_2)$ can be determined by

$$\boldsymbol{r} \cdot (\boldsymbol{a}_l^* * \boldsymbol{b}_{7-\rho-l+1}^*) + \boldsymbol{e} \cdot (\boldsymbol{a}_l^* * \boldsymbol{b}_{7-\rho-l+1}^*) + \beta_0 S(\boldsymbol{c}).$$

Since $\boldsymbol{r} \cdot (\boldsymbol{a}_l^* * \boldsymbol{b}_{7-\rho-l+1}^*)$ can be computed and $S_0(\boldsymbol{c})$ has only two possibilities, the only thing left for computing $PT_i(\boldsymbol{c}_2)$ is $\boldsymbol{e} \cdot (\boldsymbol{a}_l^* * \boldsymbol{b}_{7-\rho-l+1}^*)$. This may not be computed directly. However, as explained in the next subsection, one may obtain a finite set of values corresponding to this value and most of them is equal to $\boldsymbol{e} \cdot (\boldsymbol{a}_l^* * \boldsymbol{b}_{7-\rho-l+1}^*)$.

C. Majority voting

Let $\{\boldsymbol{a}_u^*\}$ and $\{\boldsymbol{b}_v^*\}$ as constructed in the last subsection, be two sequences satisfying the conditions (i), (ii) and (iii). For every $1 \leq u \leq 7-\rho-1$, let

$$S_{u,v} := \boldsymbol{e} \cdot (\boldsymbol{a}_u^* * \boldsymbol{b}_v^*), \ v \leq 7-\rho-u+1.$$

Then for $u+v \leq 7-\rho$, we have

$$S_{u,v} = \boldsymbol{r} \cdot (\boldsymbol{a}_u^* * \boldsymbol{b}_v^*) + \beta_{u,v,0} S_0(\boldsymbol{c})$$

which can be computed from $\boldsymbol{r}, \beta_0$ and $S_0(\boldsymbol{c})$. Therefore, we call $S_{u,v}$ known syndromes. For $1 \leq u \leq 7-\rho$, there exist $i_u \in \tilde{R}$ such that

$$\boldsymbol{a}_u^* * \boldsymbol{b}_{8-\rho-u}^* = v(\alpha^{i_u}) + \sum_{j \in R \cup \{0\}} \beta_{u,j} v(\alpha^j).$$

Thus, $S_{u,8-\rho-u}$ cannot be computed from $\boldsymbol{r}$ and are called unknowns. Consider the following matrix

$$\begin{bmatrix} S_{1,1} & S_{1,2} & \cdots & S_{1,7-\rho-2} & S_{1,7-\rho-1} & S_{1,7-\rho} \\ S_{2,1} & S_{2,2} & \cdots & S_{2,7-\rho-2} & S_{2,7-\rho-1} & - \\ \vdots & & \ddots & & \vdots & \\ S_{7-\rho-1,1} & S_{7-\rho-1,2} & \cdots & - & - & - \\ S_{7-\rho,1} & - & \cdots & - & - & - \end{bmatrix}$$

By the computation procedure in [2, 3], some of $S_{u,8-\rho-u}$, the elements on the minor diagonal, can be estimated and the majority of them have true values. For each estimated value $S_{u,8-\rho-u}^*$ of $S_{u,8-\rho-u}$, define

$$y_u = \boldsymbol{r} \cdot (\boldsymbol{a}_u^* * \boldsymbol{b}_{8-\rho-u}^*) + S_{u,8-\rho-u}^* + \beta_{u,0} S_0(\boldsymbol{c})$$

and

$$\boldsymbol{y}_u := (y_u, y_u^2, \ldots, y_u^{2^{10}}).$$

Then, the majority of $PT_{i_u}(\boldsymbol{y}_u)$ can determine $\boldsymbol{c}_2$.

D. Decoding procedure

The error-and-erasure decoding procedure for $\mathcal{G}_{23}$ up to its actual minimum distance can now be summarized as follows.

Step 1 Input the received vector $\boldsymbol{r}$ with erasure locations $k_1, \ldots, k_\rho$;

Step 2 From $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_7$ and $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_7$ and the erasure locations, find $\boldsymbol{a}_1^*, \ldots, \boldsymbol{a}_{7-\rho}^*$ and $\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_{7-\rho}^*$ such that, the conditions (i), (ii) and (iii) listed in Subsection B are satisfied;

Step 3 Let $S_0(\boldsymbol{c}) := 0$;

Step 4 Invoke the majority voting scheme and partial transform to obtain $\boldsymbol{c}_2$. Let $\boldsymbol{r}_1 := \boldsymbol{r} - \boldsymbol{c}_2$;

Step 5 Let $\boldsymbol{c} := \boldsymbol{c}_2$ if $\mathrm{wt}(\boldsymbol{r}_1) < n - \mathrm{wt}(\boldsymbol{r}_1)$. Otherwise $\boldsymbol{c} := \boldsymbol{c}_2 + (111\cdots1)$;

Step 6 If $\sum_{i=1}^{23} c_i = S_0(\boldsymbol{c})$ then output $\boldsymbol{c}$ and stop; else, let $S_0(\boldsymbol{c}) := 1$ and goto step 4.

III Error-and-erasure decoding of $\mathcal{G}_{24}$

We now consider error-and-erasure decoding of $\mathcal{G}_{24}$. Since $\mathcal{G}_{24}$ is an extended code of the quadratic residue code $\mathcal{G}_{23}$, deleting any coordinate from the codewords of $\mathcal{G}_{24}$ will result in a punctured code which is equivalent to $\mathcal{G}_{23}$ [4]. Thus for $i \in \{0, 1, \ldots, 22, \infty\}$, there exists a permutation π_i on $\{0, 1, 2, \ldots, 22\}$, such that by defining

$$\boldsymbol{c}^{(i)} := (c_0^{(i)}, \ldots, c_{22}^{(i)}) = (c_0, \ldots, c_{i-1}, c_{i+1}, \ldots, c_\infty)$$

we have

$$\pi_i(\boldsymbol{c}^{(i)}) = (c^{(i)}_{\pi_i(0)}, \ldots, c^{(i)}_{\pi_i(22)}) \in \mathcal{G}_{23}.$$

In the following, we define

$$\pi_i(\boldsymbol{x}) = (x_{\pi_i(0)}, \ldots, x_{\pi_i(22)}), \quad \text{for } \boldsymbol{x} \in \mathbf{F}_2^{23}$$

and let $\boldsymbol{y}^{(i)}$ to be a vector obtained by deleting y_i from $\boldsymbol{y} = (y_0, \ldots, y_{22}, y_\infty) \in \mathbf{F}_2^{24}$.

We now give the following error-and-erasure decoding algorithm for $\mathcal{G}_{24}$ up to its actual minimum distance.

Algorithm

Step 1 Input a received vector $\boldsymbol{r} = (r_0, \ldots, r_{22}, r_\infty)$;

Step 2 If there is no erasure, let $i := \infty$;

Else, choose any erasure location k and let $i := k$;

Step 3 Let $\boldsymbol{r}^* := \pi_i(\boldsymbol{r}^{(i)})$;

Step 4 For $\boldsymbol{r}^*$, invoke the error-and-erasure decoding algorithm of $\mathcal{G}_{23}$ to obtain $\boldsymbol{c}^*$;

Step 5 Let $\boldsymbol{c} = (c_0, \ldots, c_{23})$ such that $\boldsymbol{c}^{(i)} = \pi_i^{-1}(\boldsymbol{c}^*)$ and $c_i = \sum_{j \neq i} c_j$. Then output $\boldsymbol{c}$ and stop.

References

[1] G.D. Forney, Jr., "A bounded-distance decoding algorithm for the Leech lattice, with generalizations," *IEEE Trans. Inform. Theory*, vol. IT-35, pp. 906-909, July/August 1989.

[2] B.-Z. Shen and K. K. Tzeng, " A Code Decomposition Approach for Decoding Cyclic and Algebraic-Geometric Codes," to be presented at the 1994 International Symposium on Information Theory & Its Applications, Sydney, Australia, Nov. 20-25 1994, submitted to *IEEE Trans. Inform. Theory.*

[3] G.L. Feng and K.K. Tzeng, "A new procedure for decoding cyclic and BCH codes up to actual minimum distance," to appear in *IEEE Trans. Inform. Theory*, Sept. 1994.

[4] F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error-Correcting Codes,* Amsterdam, New York, Oxford: North Holland, 1977.

Efficient Suboptimal Decoding of Linear Block Codes[1]

Yunghsiang S. Han
HuaFan College of Humanities and Technology, Taiwan.

Ching-Tai Chiu
Carlos R. P. Hartmann
Chilukuri K. Mohan
School of Computer and Information Science, Syracuse University

Abstract

In this paper we present an efficient suboptimal soft-decision decoding algorithm for linear block codes. This decoding algorithm employs a generalization of Dijkstra's algorithm to search through a trellis for a code equivalent to the transmitted code. The use of this priority-first search strategy for decoding drastically reduces the search space. Simulation results for a sequential and parallel version of our decoding algorithm are presented.

1 Introduction

The use of block codes is a well-known error-control technique for reliable transmission of digital information over noisy communication channels. Linear block codes with good coding gains have been known for many years; however, these block codes have not been used in practice for lack of an efficient soft-decision decoding algorithm.

Several researchers [1, 14, 11] have presented techniques for decoding linear block codes that convert the decoding problem into a graph-search problem on a trellis derived from the parity-check matrix of the code. Thus the *maximum-likelihood decoding* (MLD) rule can be implemented by applying the Viterbi Algorithm [13] to this trellis. In practice, however, this breadth-first search scheme can be applied only to codes with small redundancy or to codes with a small number of codewords [9].

We recently proposed a novel maximum-likelihood soft-decision decoding algorithm for linear block codes [3, 4]. This algorithm uses a generalization of Dijkstra's algorithm (GDA) [10] to search through the trellis for a code equivalent to the transmitted code. The use of this priority-first search strategy for decoding drastically reduces the search space. In [5, 6] we showed that this decoding algorithm is very efficient for codes of moderate lengths for most practical communication systems where probability of error is less than 10^{-3}; however, for noisy channels the storage requirements of the algorithm make it impractical.

In this paper we present a suboptimal version of our decoding algorithm that can be used in practice even for very noisy channels. For the cases we simulated, only a small degradation of performance occurred. We also have implemented a parallel version of our decoding algorithm, which achieved a very good speedup for the cases we simulated.

In Section 2 we review MLD of linear block codes, describe the code tree for a linear code, and briefly state the decoding algorithm proposed in [4]. In Section 3 we present a suboptimal decoding algorithm. Simulation results for the (104,52) binary extended quadratic residue code and the (128,64) binary extended BCH code are given in Section 4. In Section 5 we give results obtained from the parallelization of GDA. Concluding remarks are presented in Section 6.

[1]This work was partially supported by the National Science Foundation under Grant NCR-9205422. It used the computational facilities of the Northeast Parallel Architectures Center (NPAC) at Syracuse University.

2 Preliminaries

Let C be a binary (n, k) linear code with generator matrix G, and let $c = (c_0, c_1, \ldots, c_{n-1})$ be a codeword of C transmitted over a time-discrete memoryless channel with output alphabet B. Furthermore, let $r = (r_0, r_1, \ldots, r_{n-1})$, $r_j \in B$ denote the received vector, and assume that $Pr(r_j|c_i) > 0$ for $r_j \in B$ and $c_i \in GF(2)$. Let $\hat{c}$ be an estimate of the transmitted codeword c.

The *maximum-likelihood decoding rule* (MLD rule) for a time-discrete memoryless channel can be formulated as [7]: set $\hat{c} = c_\ell$, where $c_\ell \in C$, and $\sum_{j=0}^{n-1} (\phi_j - (-1)^{c_{\ell j}})^2 \leq \sum_{j=0}^{n-1} (\phi_j - (-1)^{c_{ij}})^2$ for all $c_i \in C$, where $\phi_j = \ln \frac{Pr(r_j|0)}{Pr(r_j|1)}$. We, therefore, may consider that the "received vector" is $\phi = (\phi_0, \phi_1, \ldots, \phi_{n-1})$.

In the special case where the codewords of C have equal probability of being transmitted, the MLD rule minimizes error probability. We now give a short description of our decoding algorithm presented in [4]. This algorithm uses the priority-first search strategy, thus avoiding traversing the entire trellis, and, guided by an evaluation function f, searches through a graph that is a trellis for a code C^*, which is equivalent to code C. C^* is obtained from C by permuting the positions of codewords of C in such a way that the first k positions of codewords in C^* correspond to the "most reliable linearly independent" positions in the received vector ϕ. G^* is a generator matrix of C^* whose first k column forms a $k \times k$ identity matrix. In our decoding algorithm the vector $\phi^* = (\phi_0^*, \phi_1^*, \ldots, \phi_{n-1}^*)$ is used as the "received vector." It is obtained by permuting the positions of ϕ in the same manner in which the columns of G can be permuted to obtain G^*.

Since the probability that our decoding algorithm will revisit a node of the trellis is very small, our implementation of this decoding algorithm did not check for repeated nodes [4]. In this case, the graph where the search is performed is an expanded version of a trellis that is denoted by a code tree. A code tree is a way to represent every codeword of an (n, k) code C^* as a path through a tree containing $n + 1$ levels . The code tree can be treated as an expanded version of the trellis, where every path is totally distinct from every other path. The leftmost node is called the *start node*, which is at level -1. There are two branches, labeled by 0 and 1, respectively, that leave each node at the first k levels. After the k levels, there is only one branch leaving each node. The 2^k rightmost nodes are called *goal nodes*, which are at level $n - 1$.

Next, we describe how to determine the sequence of labels encountered when traversing a path from a node at level k to a goal node. Let G^* be a generating matrix of C^* whose first k columns form the $k \times k$ identity matrix. Furthermore, let $c_0, c_1, \ldots, c_{k-1}$ be the sequence of labels encountered when traversing a path from the start node to a node m at level $k - 1$. Then $c_k, c_{k+1}, \ldots, c_{n-1}$, the sequence of labels encountered when traversing a path from node m to a goal node, can be obtained as follows: $(c_0, c_1, \ldots, c_k, c_{k+1}, \ldots, c_{n-1}) = (c_0, c_1, \ldots, c_{k-1})\, G^*$.

We first specify the arc costs in the code tree of C^*. The cost of the arc from a node at level $t - 1$ to a node at level t is assigned the value $(\phi_t^* - (-1)^{c_t^*})^2$, where c_t^* is the label of the arc. Thus the solution of the decoding problem is converted into finding a path from the start node to a goal node, that is, a codeword $c^* = (c_0^*, c_1^*, \ldots, c_{n-1}^*)$ such that $\sum_{i=0}^{n-1} (\phi_i^* - (-1)^{c_i^*})^2$ is minimum among all paths from the start node to goal nodes. Such a path is denoted as an optimal path.

Now we define the evaluation function f for every node m in the code tree as $f(m) = g(m) + h(m)$, where $g(m)$ is the cost of the path from the start node to node m and $h(m)$ is an estimate of the minimum cost among all the paths from node m to goal nodes. The cost of a path is obtained by summing all the arc costs encountered while constructing this path. GDA requires that for all nodes m_i and m_j such that node m_j is an immediate successor of node m_i,

$$h(m_i) \leq h(m_j) + c(m_i, m_j), \tag{1}$$

where $c(m_i, m_j)$ is the arc cost between node m_i and node m_j. This requirement guarantees that GDA will find an optimal path. In GDA, the next node to be expanded is one with the smallest value of f on the sorted list "OPEN" containing all leaf nodes of the subtree constructed so far by the algorithm. The algorithm terminates when it chooses to expand a goal node, because it has constructed an optimal path.

We now define our function h, which satisfies (1). In order to define a function h that is a "good" estimator, we must use properties of the linear block code that are invariant under any permutation of the positions of the codewords.

Let $HW = \{w_i | 0 \leq i \leq I\}$ be the set of all distinct Hamming weights that codewords of $\boldsymbol{C}$ may have. Furthermore, assume $w_0 < w_1 < \cdots < w_I$. Our heuristic function is defined to take into consideration the linear property of $\boldsymbol{C}^*$ and that the Hamming distance between any two codewords of $\boldsymbol{C}^*$ must belong to HW.

Let $\boldsymbol{c}^*$ be a given codeword of $\boldsymbol{C}^*$, which is called the seed of the decoding algorithm.

1. For nodes at level ℓ, $-1 \leq \ell < k-1$:

   Let m be a node at level ℓ, and let $\overline{v}_0, \overline{v}_1, \ldots, \overline{v}_\ell$ be the labels of the path $\boldsymbol{P}'_m$ from the start node to node m. We now construct the set, $T(m)$, of all binary n-tuples $\boldsymbol{v}$ such that their first $\ell+1$ entries are the labels of $\boldsymbol{P}'_m$ and $d_H(\boldsymbol{v}, \boldsymbol{c}^*) \in HW$, where $d_H(\boldsymbol{x}, \boldsymbol{y})$ is the Hamming distance between $\boldsymbol{x}$ and $\boldsymbol{y}$. That is, $T(m) = \{\boldsymbol{v} | \boldsymbol{v} = (\overline{v}_0, \overline{v}_1, \ldots, \overline{v}_\ell, v_{\ell+1}, \ldots, v_{n-1})$ and $d_H(\boldsymbol{v}, \boldsymbol{c}^*) \in HW\}$. Note that $T(m) \neq \emptyset$ since $\boldsymbol{u} \cdot \boldsymbol{G}^* \in T(m)$, where $\boldsymbol{u} = (\overline{v}_0, \overline{v}_1, \ldots, \overline{v}_\ell, 0, \ldots, 0)$.

   We now define function h as $h(m) = \min_{\boldsymbol{v} \in T(m)} \left\{ \sum_{i=\ell+1}^{n-1} \left(\phi_i^* - (-1)^{v_i}\right)^2 \right\}$.

2. For nodes at level ℓ, $k-1 \leq \ell < n$:

   Let m be a node at level ℓ. We define function h as $h(m) = \sum_{i=\ell+1}^{n-1} \left(\phi_i^* - (-1)^{v_i^*}\right)^2$, where $v_{\ell+1}^*, v_{\ell+2}^*, \ldots, v_{n-1}^*$ are the labels of the only path $\boldsymbol{P}_m$ from node m to a goal node. Note that if node m is a goal node, then $h(m) = 0$.

In [3, 4] it is shown that our decoding algorithm is a depth-first search type algorithm. Thus, upper bounds (UBs) on the cost of an optimal path are obtained whenever a codeword is generated. These UBs can be used to reduce the size of list OPEN. More details about this decoding algorithm can be found in [3, 4] where we also described other speed-up techniques.

3 Suboptimal decoding algorithm

In [5, 6] we showed that GDA is very efficient for codes of moderate lengths for most practical communication systems where probability of error is less than 10^{-3}; however, for low signal to noise ratio (SNR) the number of nodes on list OPEN is still too large for the algorithm to have practical application. Simulations have shown that the number of nodes that need to be stored on list OPEN before finding an optimal path is considerably smaller than the total number of nodes stored before the algorithm stops. Thus we may limit the search with small degradations in the performance of the algorithm.

In this section we present a suboptimal soft-decision decoding algorithm in which we limit the size of list OPEN using two criteria that we will describe next.

1. If a node m needs to be stored on list OPEN when the size of list OPEN has reached a given upper bound, then we discard the node with larger f value between node m and the node on list OPEN with the maximum value of function f.

2. If the probability that an optimal path goes through a node is smaller than a given parameter, then we do not store this node.

To use the second criterion we need to calculate the probability that an optimal path goes through a node. For any received vector ϕ^*, if an optimal decoding algorithm decodes it to a non-transmitted codeword, then it is very difficult for a suboptimal decoding algorithm to decode it to the transmitted codeword. Thus, when an optimal decoding algorithm decodes a received vector to a non-transmitted codeword we do not care which codeword a suboptimal decoding algorithm will decode to. Therefore, it is reasonable to consider only those received vectors that will be decoded to transmitted codewords by an optimal decoding algorithm. That is, when we derive the probability that an optimal path goes through a node, we will assume that no decoding error will occur if we employ an optimal decoding algorithm. Let $h^*(m_s)$ note the cost of an optimal path. Under this assumption we have the following theorem.

Let an (n, k) code $\boldsymbol{C}$ be transmitted over the AWGN channel. When no decoding error occurs, the probability distribution of $h^*(m_s)$ is approximately a normal distribution with mean $n\frac{N_0}{2}$ and variance $n\frac{N_0^2}{2}$.

The proof of the above result is given in the Appendix.

Let node m be a node in the code tree of the transmitted (n, k) code $\boldsymbol{C}$ and let UB be the lowest upper bound on the cost of an optimal path found so far by the algorithm. In [4] it is shown that for any node m of the code tree, $h(m) \leq h^*(m)$ where $h^*(m)$ is the actual cost of the minimum cost path among all the paths from node m to goal nodes. Thus, if an optimal path goes through node m, then $h(m) \leq h^*(m) \leq h^*(m_s)$, and the probability that an optimal path goes through node m is less than or equal to $\boldsymbol{Pr}(h(m) \leq h^*(m_s) \leq UB)$. If we let T be the probability that an optimal path goes through node m and UB be an upper bound on the cost of an optimal path, then $T \leq T_{UB}$, where $T_{UB} = \frac{1}{\sigma\sqrt{2\pi}} \int_{h(m)}^{UB} e^{-\frac{1}{2}(\frac{t-\mu}{\sigma})^2} dt$, $\mu = n\frac{N_0}{2}$, and $\sigma^2 = n\frac{N_0^2}{2}$. Thus, when a node is visited, the algorithm calculates T_{UB} for this node. If this value is less than a given threshold, then we will discard this node.

In our suboptimal decoding algorithm we fix the maximum number of nodes, M_B, allowed on list OPEN. As in an optimal decoding algorithm, list OPEN is always kept ordered. When a node m is visited, the algorithm calculates T_{UB} for this node. If T_{UB} is less than a given threshold δ, then discard this node. Otherwise, we need to insert this node into list OPEN. If the number of nodes on list OPEN is equal to M_B, then the algorithm discards the node with larger f value between node m and the node with the largest f value on list OPEN. The algorithm inserts the remaining node into list OPEN. We remark here that all the speed-up techniques in [4] can be applied to the suboptimal decoding algorithm.

4 Simulation results for the AWGN channel

In order to verify the performance of our suboptimal decoding algorithm, we present simulation results for the (104, 52) binary extended quadratic residue code and the (128, 64) binary extended BCH code when these codes are transmitted over an Additive White Gaussian Noise (AWGN) channel. We assume that antipodal signaling is used in the transmission, so that the j^{th} components of the transmitted codeword $\boldsymbol{c}$ and received vector $\boldsymbol{r}$ are $c_j = (-1)^{c_j}\sqrt{E}$ and $r_j = (-1)^{c_j}\sqrt{E} + e_j$, respectively, where E is the signal energy per channel bit and e_j is a noise sample of a Gaussian process with single-sided noise power per hertz N_0. The variance of e_j is $N_0/2$ and the SNR for the channel is $\gamma = E/N_0$. In order to account for the redundancy in codes of different rates, we used the SNR per transmitted information bit $\gamma_b = E_b/N_0 = \gamma n/k$ in our simulation. For the AWGN channel, $\boldsymbol{\phi} = \frac{4\sqrt{E}}{N_0}\boldsymbol{r}$ [7], so we can substitute $\boldsymbol{r}(\boldsymbol{r}^*)$ for $\boldsymbol{\phi}(\boldsymbol{\phi}^*)$ in our decoding algorithm.

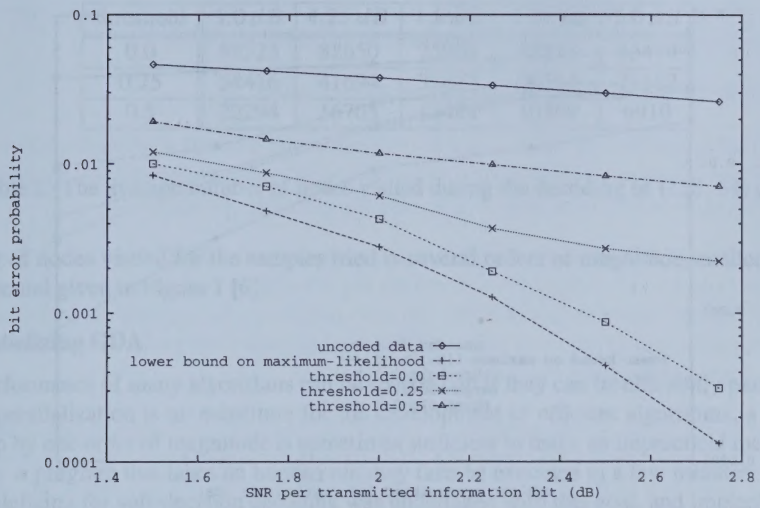


Figure 1: Performance of suboptimal decoding algorithm for the (104, 52) code

threshold	1.5 dB	1.75 dB	2.0 dB	2.25 dB	2.5 dB	2.75 dB
0.0	26357	23909	18366	13240	10070	6698
0.25	10976	9643	6481	3980	2879	1579
0.5	3166	2827	1818	950	703	344

Table 1: The average number of nodes visited during the decoding of (104, 52) code

We do not know HW for these two codes, so we use a superset for them. For (104,52) we know that $d_{\min} = 20$ and that the Hamming weight of any codeword is divisible by 4 [8]. Thus for this code the superset used is $\{x|(x$ is divisible by 4 and $20 \leq x \leq 84)$ or $(x = 0)$ or $(x = 104)\}$; for (128,64), the superset used is $\{x|$ (x is even and $22 \leq x \leq 106)$ or $(x = 0)$ or $(x = 128)\}$, since this code has $d_{\min} = 22$.

We have implemented a suboptimal version of the adaptive decoding algorithm presented in [4]. For low SNR, the initial $\boldsymbol{c}_0^*$ is constructed by considering the 16 codewords as follows. Let $\boldsymbol{y} = (y_0, y_1, \ldots, y_{n-1})$ be the hard-decision of $\boldsymbol{r}^*$. Furthermore, let $S = \{\, \boldsymbol{u} | \, \boldsymbol{u} = (u_0, u_1, \ldots, u_{k-1})$ and $u_i = y_i$ for $0 \leq i \leq k-5\}$. For every element $\boldsymbol{u}$ in S, we get a codeword $\boldsymbol{c}^* = \boldsymbol{u} \cdot \boldsymbol{G}^*$. Now we let $\boldsymbol{c}_0^* = \boldsymbol{c}^*$, where the value of $h(m_s)$, calculated with respect to $\boldsymbol{c}^*$, is the largest among all the 16 codewords. The rule for updating the seed is the same as that in [4].

The simulation results for the (104, 52) code for γ_b equal to 1.5 dB, 1.75 dB, 2.0 dB, 2.25 dB, 2.5 dB, and 2.75 dB are given in Figure 1 and in Table 1 for three threshold values for the case when M_B is equal to 3000.

In Figure 1 we also give a lower bound on the bit error probability of the maximum-likelihood decoding algorithm. This lower bound is obtained as follows [2]. For every sample, when the

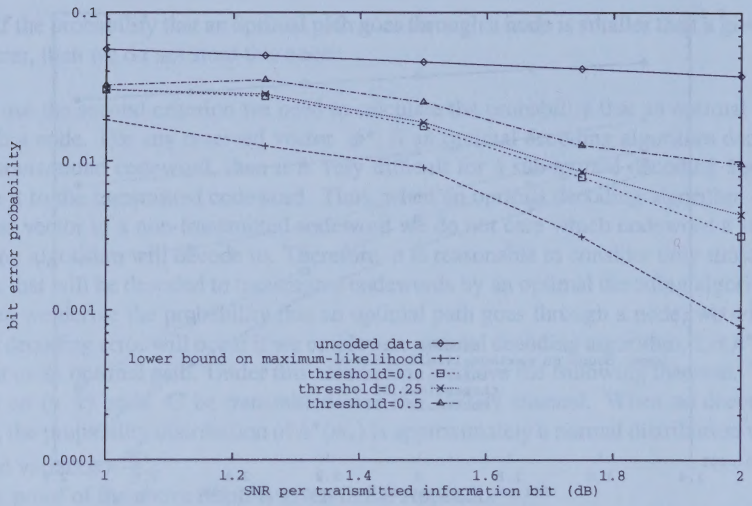


Figure 2: Performance of suboptimal decoding algorithm for the (128, 64) code

suboptimal decoding algorithm terminates, we have a codeword that is obtained from the algorithm. If this codeword is closer with respect to Euclidean distance to the received vector than the transmitted codeword, then any optimal decoding algorithm will also decode the received vector to a non-transmitted codeword. Thus, we assume that the optimal decoding algorithm will decode to the codeword obtained from the suboptimal decoding algorithm and report a decoding error occurs. Bit error probability of the uncoded data is also given in Figure 1.

From Figure 1, for the (104, 52) code the performance of the suboptimal decoding algorithm with $\delta = 0.0$ is within 0.25 dB of the performance of an optimal decoding algorithm; the performance of the suboptimal decoding algorithm with $\delta = 0.25$ is within 0.5 dB of the performance of an optimal decoding algorithm; and the performance of the suboptimal decoding algorithm with $\delta = 0.5$ is within 1 dB of the performance of an optimal decoding algorithm. Thus, for the samples tried, limiting the size of list OPEN to 3000 nodes introduced only a small degradation on the performance of the algorithm for the (104, 52) code. However, the average number of nodes visited for the sample tried is several orders of magnitude smaller than the upper bound given in Figure 1 [6].

The simulation results for the (128, 64) code for γ_b equal to 1.0 dB, 1.25 dB, 1.5 dB, 1.75 dB, and 2.0 dB are given in Figure 2 and in Table 2 for three threshold values for the case when M_B is equal to 6000. In Figure 2 we also give a lower bound on the bit error probability of the maximum-likelihood decoding algorithm and bit error probability of the uncoded data.

From Figure 2, for the (128, 64) code the performance of the suboptimal decoding algorithm with $\delta = 0.0$ is within 0.5 dB of the performance of an optimal decoding algorithm; the performance of the suboptimal decoding algorithm with $\delta = 0.25$ is within 0.6 dB of the performance of an optimal decoding algorithm; and the performance of the suboptimal decoding algorithm with $\delta = 0.5$ is within 0.75 dB of the performance of an optimal decoding algorithm. Thus, for the samples tried, limiting the size of list OPEN to 6000 nodes introduced only a small degradation on the performance of the algorithm for the (128, 64) code. However, the average

threshold	1.0 dB	1.25 dB	1.5 dB	1.75 dB	2.0 dB
0.0	88325	82650	75905	65223	55474
0.25	54416	41694	35613	29554	23162
0.5	22294	16705	13478	10389	6910

Table 2: The average number of nodes visited during the decoding of (128, 64) code

number of nodes visited for the samples tried is several orders of magnitude smaller than the upper bound given in Figure 1 [6].

5 Parallelizing GDA

The performance of many algorithms can be speeded up if they can be efficiently parallelized. While parallelization is no substitute for the development of efficient algorithms, a run-time speedup by one order of magnitude is sometimes sufficient to make an impractical method reasonable: a program that takes an hour to run may then be executed in a few minutes. The task of parallelizing for soft-decision decoding was undertaken with this goal, and implemented on a massively parallel computer, the CM-5.[2] The main bottleneck in this task is the fact that there is a global OPEN list that needs to be accessed by each processor.[3]

We implemented GDA on a CM-5, using message-passing communication. In the sequential version, an OPEN list is maintained, sorted in heuristic order.[4] In the parallel version, we distributed the nodes in the OPEN list equally to all processors. All processors concurrently expand the best nodes in their local OPEN lists. New nodes are generated as a result, and need to be added to the OPEN list of some processor. Since some nodes generate more children than others, severe load imbalances can result if we add the newly generated nodes to the OPEN list of the same processor which led to their generation; some processors may then be idle whereas others have excessively large OPEN lists. Hence, newly generated nodes are distributed to other processors preserving the property that each local OPEN list is roughly of the same size. The termination criterion for the parallel algorithm is similar to that of the sequential algorithm: the procedure terminates when either all local OPEN lists are empty or an acceptable solution is found by some processor.

The following data structures are used by the parallel algorithm.

1. Each processor has a *local OPEN list* to store the opened nodes in heuristic order. Because of memory limitations, the maximum possible size of all OPEN lists is predetermined. If the global memory bound for all OPEN lists is M_B, and i is the number of processors, then the size of each local OPEN list is at most M_B/i.

2. Each processor has a *WAITING list* to store the expanded nodes waiting to be distributed to various processors; these nodes arise as a result of expanding a node from the local OPEN list.

3. A *Global upper bound* is used to trim the size of OPEN list to save execution time; this is similar to the upper bound used in the sequential algorithm. For each "expanding cycle"

[2]CM-5 is a recent model of the Connection Machine, manufactured by Thinking Machines, Inc.

[3]Notation: A "node" is a vertex in the code tree, whereas a "processor" is a computing element in a parallel computer or multiprocessor.

[4]Recall that the OPEN list stores all the opened nodes waiting to be expanded.

(the period between distributing successive nodes opened at a processor), a new *local upper bound* may be generated and broadcast to other processors.

Two kinds of communications occur between processors:

Broadcast new upper bound: When a new local upper bound is found, it will be broadcast to all other processors to update the global upper bound. If several different local upper bounds are found, the smallest of them will replace the global upper bound.

Node distribution: The algorithm exploits the ring topology of the CM-5, in which each processor is connected to two others. There are two alternating cycles, in which each processor distributes nodes in its WAITING list to its neighbors. In each odd cycle, a processor distributes nodes in the clockwise direction, and in each even cycle, the WAITING list nodes are distributed in counter-clockwise direction.

In order to verify the performance of our parallel version of our suboptimal decoding algorithm, we present simulation results for the (104, 52) binary extended quadratic residue code transmitted over the AWGN channel using antipodal signaling. In this simulation M_B was set to 12000 and δ to 0.0.

In Figure 3 we give the speedup obtained for various values of i for γ_b equal to 1.0 dB and 1.25 dB. For these cases we obtained a very good speedup. Figure 3 also shows that no substantial degradation in the performance of the algorithm occurred, because each processor could store at most M_B/i nodes. In Figure 3 we also give a lower bound on the bit error probability of the maximum-liklihood decoding algorithm.

6 Conclusions

For low SNR we propose a suboptimal decoding algorithm. From Figures 1 and 2, for the SNR we tried, the performance of this algorithm is within 0.25 dB of the performance of an optimal decoding algorithm for the (104, 52) binary extended quadratic residue code and within 0.5 dB for the (128, 64) binary extended BCH code.

We have also implemented a parallel version of our decoding algorithm. As Figure 3 shows, we obtained a very good speedup for the cases we tried. Furthermore, almost no degradation in performance occurred due to the fact that the memory of each processor is M_B/i.

Appendix

Let an (n, k) code $\boldsymbol{C}$ be transmitted over an AWGN channel. If we assume that no decoding error occurs, $f^*(m_s) = \sum_{i=0}^{n-1} \left((-1)^{c_i}\sqrt{E} - (e_i + (-1)^{c_i}\sqrt{E})\right)^2 = \sum_{i=0}^{n-1} e_i^2$, where for each $0 \leq i \leq n-1$, e_i's are $i.i.d.$ and e_i is a normal random variable with mean 0 and variance $N_0/2$. Consequently, $\frac{2}{N_0}\sum_{i=0}^{n-1} e_i^2 = \frac{2}{N_0} f^*(m_s)$ will be distributed as a chi-square random variable with n degrees of freedom. From [12] it follows that $E(f^*(m_s)) = n\frac{N_0}{2}$, and $\mathrm{Var}(f^*(m_s)) = n\frac{N_0^2}{2}$. However, for large values of n, the probability distribution of $f^*(m_s)$ is approximately a normal distribution with mean μ and variance σ^2, given above.

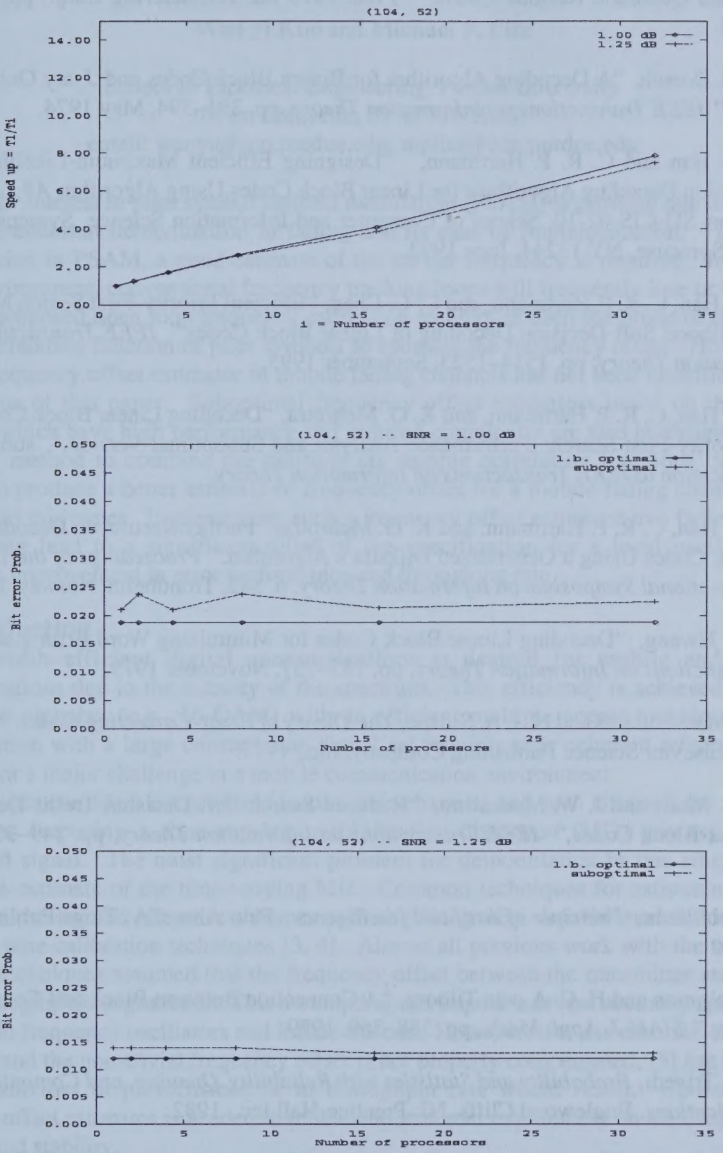


Figure 3: Performance of parallel suboptimal decoding algorithm for the (104,52) code

References

[1] R.W. D. Booth, M. A. Herro, and G. Solomon, "Convolutional Coding Techniques for Certain Quadratic Residue Codes," *Proc. 1975 Int. Telemetering Conf.*, pp. 168–177, 1975.

[2] B. G. Dorsch, "A Decoding Algorithm for Binary Block Codes and J-ary Output Channels," *IEEE Transactions on Information Theory*, pp. 391–394, May 1974.

[3] Y. S. Han and C. R. P. Hartmann, "Designing Efficient Maximum-Likelihood Soft-Decision Decoding Algorithms for Linear Block Codes Using Algorithm A^*," Technical Report SU-CIS-92-10, School of Computer and Information Science, Syracuse University, Syracuse, NY 13244, June 1992.

[4] Y. S. Han, C. R. P. Hartmann, and C.-C. Chen, "Efficient Priority-First Search Maximum-Likelihood Soft-Decision Decoding of Linear Block Codes," *IEEE Transactions on Information Theory*, pp. 1514–1523, September 1993.

[5] Y. S. Han, C. R. P. Hartmann, and K. G. Mehrotra, "Decoding Linear Block Codes Using a Priority-First Search: Performance Analysis and Suboptimal Version," submitted for publication to *IEEE Transactions on Information Theory*.

[6] Y. S. Han, C. R. P. Hartmann, and K. G. Mehrotra, "Further Results on Decoding Linear Block Codes Using a Generalized Dijkstra's Algorithm," *Proceedings of the 1994 IEEE International Symposium on Information Theory*, p. 342, Trondheim, Norway, June 1994.

[7] T.-Y. Hwang, "Decoding Linear Block Codes for Minimizing Word Error Rate," *IEEE Transactions on Information Theory*, pp. 733–737, November 1979.

[8] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. New York, NY: Elsevier Science Publishing Company, Inc., 1977.

[9] K. R. Matis and J. W. Modestino, "Reduced-Search Soft-Decision Trellis Decoding of Linear Block Codes," *IEEE Transactions on Information Theory*, pp. 349–355, March 1982.

[10] N. J. Nilsson, *Principle of Artificial Intelligence*. Palo Alto, CA: Tioga Publishing Co., 1980.

[11] G. Solomon and H. C. A. van Tilborg, "A Connection Between Block and Convolutional Codes," *SIAM J. Appl. Math.*, pp. 358–369, 1979.

[12] K. S. Trivedi, *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*. Englewood Cliffs, NJ: Prentice-Hall Inc., 1982.

[13] A. J. Viterbi, "Error Bound for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Transactions on Information Theory*, pp. 260–269, April 1967.

[14] J. K. Wolf, "Efficient Maximum Likelihood Decoding of Linear Block Codes Using a Trellis," *IEEE Transactions on Information Theory*, pp. 76–80, January 1978.

Frequency Offset Compensation in Frequency Flat Fading*

Wen-yi Kuo and **Michael P. Fitz**

School of Electrical Engineering, Purdue University
West Lafayette, IN 47907-1285
email: wenyi@ecn.purdue.edu, mpfitz@ecn.purdue.edu

Abstract: Interest in pilot symbol assisted modulation (PSAM) is growing due to its ability to provide coherent demodulation in fading and its ease of implementation. For optimal demodulation in PSAM, a good estimate of the carrier frequency is required. In a mobile fading environment, conventional frequency tracking loops will frequently lose or false lock. Recently proposed open loop frequency estimation techniques used the phase of the one-lag sample correlation function of pilot symbols to estimate the frequency offset. However, the optimal frequency offset estimator in mobile fading channels has not been identified and is a major focus of this paper. Suboptimal frequency offset estimators based on the optimal estimator which have high performance and reduced complexity are also investigated in this paper. A method to combine the multiple lag sample correlation functions of the pilot symbols to produce a better estimate of frequency offset for a mobile fading channel is also presented in this paper. Implementing such a frequency offset estimator in a fading channel receiver can lead to a significant relief of the specification for a local oscillator or a significant improvement in error performance and throughput rate.

1. Introduction

Bandwidth efficient digital communications is desired for mobile and personal communications due to the scarcity of the spectrum. This efficiency is achieved if a large modulation alphabet (e.g., 16-QAM) with an efficient multiple access technique is used. Demodulation with a large constellation signal set requires near coherent reference which accounts for a major challenge in a mobile communication environment.

This paper considers *narrowband* land mobile channels and such channels are accurately modeled as imposing a time-varying multiplicative distortion (MD) process upon the transmitted signal. The most significant problem for demodulation in this situation is to produce an estimate of the time-varying MD. Common techniques for estimating the MD process are transmitted reference schemes such as pilot symbol assisted modulation (PSAM) [1, 2] and tone calibration techniques [3, 4]. Almost all previous work with the transmitted reference techniques assumed that the frequency offset between the transmitter and receiver produced negligible degradation. This assumption can impose a severe accuracy specification on the radio frequency oscillators and inflate the cost. However, if a less accurate oscillator is employed and the non-trivial frequency offset is not properly compensated, [5] has found that a significant loss in performance or in throughput rate would result. Consequently a frequency offset estimator is needed to relieve the stringent requirement on the oscillator's set accuracy and stability.

This paper addresses the issue of frequency offset estimation for the PSAM in wireless mobile digital communications. The traditional method of frequency estimation is to employ

* This work partially supported by the National Science Foundation under Grant NCR-9115820.

an automatic frequency tracking loop [6-9] in the receiver. However, signals passing through the wireless channel experience fading distortion on both the amplitude and the phase. From time to time the multipath channel characteristics produce the deep fade where the amplitude is severely attenuated and the phase changes rapidly. Conventional frequency tracking loops will thus suffer a larger estimation error variance, an error floor in bit error probability (BEP), and a longer acquisition time in this fading situation. Some recent efforts [10-14] utilize the correlations of transmitted references to estimate the frequency offset. These algorithms either are designed for non-fading AWGN channels or only use the one-step correlation of transmitted references for mobile fading channels. None of the previous research has studied the optimal estimation of frequency offset in a mobile fading environment nor has it investigated using multiple lags of the sample correlation function of transmitted reference signal to estimate the frequency offset. These two areas are the major focuses of this paper.

This paper is organized as follows. Section 2 introduces the analytical models. Section 3 studies the maximum likelihood frequency offset estimator and suboptimal approximations. Section 4 investigates the reduced complexity frequency offset estimator based on the sample autocorrelation function of the received signals (i.e., the pilot symbols). Section 5 concludes.

2. Analytical Models

Fig. 1 is a block diagram of the analytical model for a PSAM based communication system. Linear modulations are considered and the transmitted signal has the form

$$s(t) = \sum_i d_i u(t - iT) \tag{1}$$

where $u(t)$ represents a square root unit energy Nyquist pulse shape[1] and T is the symbol duration. The complex valued d_i's are the transmitted modulation symbols and are taken from an MPSK or M-QAM alphabet where the d_i's are normalized such that $E\left\{\left|d_i\right|^2\right\} = \log_2 M$. The known pilot symbols of PSAM are inserted in the transmitted symbols to help the demodulator derive a channel state estimate.

The channel model for this paper is the frequency non-selective, time-varying, isotropic scattering, Rayleigh fading channel with a possible frequency offset between the transmitter and receiver. The input to the receiver is then given by

$$z(t) = e^{j2\pi f_o t} c(t) s(t) + n(t) \tag{2}$$

where $c(t)$ is a complex Gaussian multiplicative distortion (MD), f_o is a random variable representing the unknown frequency offset, and $n(t)$ is a complex AWGN process with variance N_0. Note that $n(t)$, $c(t)$, and f_o are jointly and individually independent of each other. The isotropic Rayleigh scattering assumption implies that $c(t)$ is of zero mean and characterized by the Fourier transform pair [15]

$$R_c(\tau) = E\left\{c(t)c^*(t-\tau)\right\} = \overline{E}_b J_0(2\pi f_D \tau) \qquad S_c(f) = \frac{\overline{E}_b}{\pi f_D}\left(1 - \frac{f^2}{f_D^2}\right)^{-\frac{1}{2}} \qquad |f| < f_D \tag{3}$$

where f_D represents the Doppler spread and $\overline{E}_b$ is the average received energy per bit.

---

[1] $R_u(nT) = \int_{-\infty}^{\infty} u(t)u^*(t-nT)dt = \delta_n$

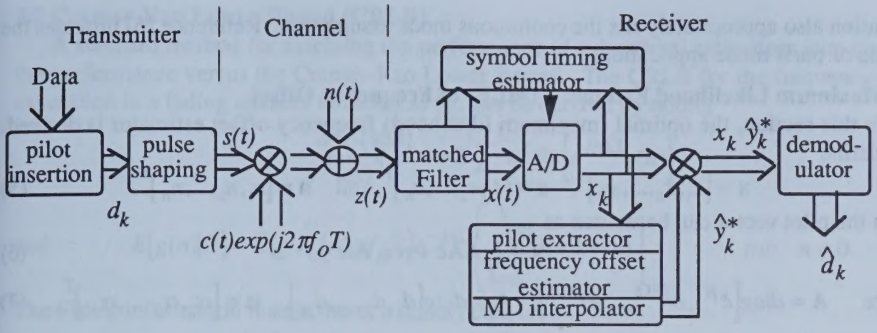


Fig. 1. Block diagram of a pilot symbol assisted communication system where y_k denotes the total MD, $c_k e^{jk\Omega}\xi_u(\Omega)$, of the signal.

The optimal demodulation structure for the signal models given in (1)-(3) is quite complicated. This paper considers a more practical architecture which employs a pulse shape matched filter and then performs the frequency estimation with the matched filter output samples. Assuming perfect symbol timing recovery and the first-order approximation that the pulse shape is narrow compared to $1/f_D$ so that the MD is approximately constant for a pulse duration and the ISI produced by the MD is negligible, the output samples of the pulse shape matched filter thus take the following form

$$x_k = \int_{-\infty}^{\infty} z(t)u^*(t-kT)dt = d_k c_k e^{j2\pi f_0 kT}\int_{-\infty}^{\infty} e^{j2\pi f_0 t}|u(t)|^2 dt + n_k = d_k c_k e^{jk\Omega}\xi_u(\Omega) + n_k \quad (4)$$

where c_k is a discrete time zero mean complex Gaussian process with $E[c_k c_{k-m}^*] = R_c(mT)$, $\Omega = 2\pi f_O T$ is the normalized frequency offset (radians/sample), n_k is a white Gaussian noise sequence with variance N_0 and $\xi_u(\Omega)$ is the amplitude loss caused by the frequency offset for the pulse shape $u(t)$. The fluctuation of $\xi_u(\Omega)$ due to Ω is insignificant so that $\xi_u(\Omega)$ is approximated as a constant, ξ_u, independent of Ω for the reminder of this paper. $\overline{E}_{eff} = \xi_u(\Omega)^2\overline{E}_b$ denotes the effective received energy per bit of the receiver.

The purpose of PSAM is to estimate the total MD, $\xi_u c_k e^{jk\Omega}$, and provide near coherent demodulation of the data symbols. Since achieving high performance requires a frequency shifted pilot symbol interpolation filter for demodulation, accurate frequency estimation is imperative [5].

3. Maximum Likelihood Based Frequency Offset Estimators

This section examines the optimal frequency offset estimation and suboptimal approximation. In this paper we consider the frequency offset estimation in continuous mode operation where N consecutive matched filter output samples with known modulation are assumed available at the demodulator to produce a frequency offset estimate. The continuous mode assumption is a general description of many applications. For instance, a system with periodic pilot insertion can be modeled in this way by simply substituting the pilot insertion period for T. A known synchronization preamble or a training sequence exactly fits the continuous mode operation model. Often the demodulation and channel tracking utilizes the decision feedback algorithms. When the decisions are of high quality, the decision feedback

operation also appropriately fits the continuous mode assumption. Reference [5] provides the details of burst mode applications.

3.1 Maximum Likelihood Estimator (MLE) of Frequency Offset

In this section, the optimal (maximum likelihood) frequency offset estimator is derived. We define

$$\mathbf{x} = [x_1, x_2, ..., x_N]^T, \quad \mathbf{c} = [c_1, c_2, ..., c_N]^T, \text{ and } \quad \mathbf{n} = [n_1, n_2, ..., n_N]^T. \tag{5}$$

Then the pilot vector can be written as

$$\mathbf{x} = \xi_u \mathbf{\Lambda A c} + \mathbf{n} = \mathbf{\Lambda \alpha} \tag{6}$$

where $$\mathbf{A} = diag\left[e^{j\Omega}, e^{j2\Omega}, ..., e^{jN\Omega}\right], \quad \mathbf{\Lambda} = diag\left[d_1, d_2, ..., d_N\right], \quad \mathbf{\alpha} = [\alpha_1, \alpha_2, ..., \alpha_N]^T, \tag{7}$$

and

$$\alpha_k = \frac{x_k}{d_k} = \xi_u c_k e^{jk\Omega} + \frac{n_k}{d_k} \tag{8}$$

is the demodulated pilot symbol (i.e., received pilot symbol with the known modulation removed). The vector $\mathbf{x}$ conditioned on Ω is now Gaussian with zero mean and covariance

$$\mathbf{R_x} = E\left[\mathbf{x}\mathbf{x}^H | \Omega\right] = \xi_u^2 \mathbf{\Lambda A} E\left[\mathbf{c}\mathbf{c}^H\right] \mathbf{A}^H \mathbf{\Lambda}^H + N_o \mathbf{I} = \mathbf{A}\left(\xi_u^2 \mathbf{\Lambda R_c \Lambda}^H + N_o \mathbf{I}\right)\mathbf{A}^H \tag{9}$$

where $\mathbf{I}$ is an $N \times N$ identity matrix and $\mathbf{R_c} = E\left[\mathbf{c}\mathbf{c}^H\right]$.

Consequently, the log-likelihood function for frequency estimation is given as

$$\Lambda(\mathbf{x}|\Omega) = C(\mathbf{x}) - \mathbf{x}^H \mathbf{R_x^{-1}} \mathbf{x} = C(\mathbf{x}) + \mathbf{\alpha}^H \mathbf{A B A}^H \mathbf{\alpha} = C(\mathbf{x}) + g(0) + 2\sum_{n=1}^{N-1} \mathrm{Re}\left[e^{jn\Omega} g(n)\right] \tag{10}$$

where

$$\mathbf{B} = -\mathbf{\Lambda}^H \left(\xi_u^2 \mathbf{\Lambda R_c \Lambda}^H + N_o \mathbf{I}\right)^{-1} \mathbf{\Lambda} = -\left(\xi_u^2 \mathbf{R_c} + N_o \mathbf{\Delta}\right)^{-1}, \tag{11}$$

$$\mathbf{\Delta} = diag\left(|d_1|^{-2}, |d_2|^{-2}, ..., |d_N|^{-2}\right), \qquad g(n) = \sum_{k=k_{mi}}^{k_{ma}} \mathbf{B}(k, k-n) \alpha_k^* \alpha_{k-n}, \tag{12}$$

$C(\mathbf{x})$ is a constant independent of Ω, $k_{mi} = \max\{1, n+1\}$, and $k_{ma} = \min\{N, N+n\}$. Notice that $\mathbf{B}$ is an inverse correlation matrix which accounts for the randomness of $\mathbf{x}$ due to the Doppler spread, the additive noise, and the pilot symbol modulations. $\mathbf{B}$ is a real and symmetric matrix (i.e., $\mathbf{B}^H = \mathbf{B} = \mathbf{B}^*$). Also note that $g(n)$ is a sample autocorrelation function with non-uniform weighting and is Hermitian symmetric, i.e., $g(-n)=g^*(n)$.

The maximum likelihood estimate of Ω can be obtained from (10) and is given as

$$\Omega_{ML} = arg \max_{\Omega} \Lambda(\mathbf{x}|\Omega). \tag{13}$$

In general, (13) is not explicitly solvable. An exhaustive search over the candidate range (the range where $f_o T$ is possibly located) can produce a near-optimal estimate but at a potentially high complexity. A common way to find an MLE is to solve the likelihood equation which is

$$0 = \frac{d\Lambda(\mathbf{x} \mid \Omega)}{d\Omega} = -2\,\mathrm{Im}\left(\sum_{n=1}^{N-1} n e^{jn\Omega} g(n)\right) = \sum_{n=1}^{N-1} n r_n \sin\left(\phi_n + n\Omega\right). \tag{14}$$

where

$$\phi_n = \arg\{g(n)\} \quad and \quad r_n = |g(n)|. \tag{15}$$

Root finding has many well investigated methods but unfortunately the likelihood function has many local maximum. A gradient type search will not produce a good estimate unless the initial starting point is within the domain of attraction of the global maximum.

3.2 Cramer-Rao Lower Bound (CRLB)

A standard method for assessing the performance of suboptimal estimators is to compare the performance versus the Cramer-Rao Lower Bound. The CRLB for the frequency offset estimation in a fading channel modeled as (6) can be derived by noting

$$\frac{d^2\Lambda(\mathbf{x}|\Omega)}{d\Omega^2} = -2\sum_{n=1}^{N-1} n^2 \,\mathrm{Re}\left[e^{jn\Omega} g(n)\right] \tag{16}$$

and $$E[g(n)|\Omega] = \overline{E}_{eff} J_0(n2\pi f_D T) e^{-jn\Omega} \left(\sum_{k=k_{mi}}^{k_{ma}} \mathbf{B}(k, k-n) \right) \quad for \quad n \neq 0. \tag{17}$$

The Fisher information matrix (here, a scalar) then is

$$J = -E\left[\frac{d^2\Lambda(\mathbf{x}\,|\,\Omega)}{d\Omega^2}\right] = 2\overline{E}_{eff} \sum_{n=1}^{N-1} \left\{ n^2 J_0(n2\pi f_D T) \sum_{k=n+1}^{N} \mathbf{B}(k, k-n) \right\} \tag{18}$$

and the CRLB for the estimation of Ω is given as $E\left[\left(\Omega - \hat{\Omega}\right)^2\right] \geq J^{-1}$.

3.3 Linear Approximation MLE

Since a global search technique for finding a ML estimate is computationally unattractive and traditional gradient search method's performance is highly dependent on the initial starting point, reduced complexity approximations to the MLE are of interest. Note in (17) that $\arg\{E[g(n)|\Omega]\} = -n\Omega$ which suggests that, for each n, ϕ_n/n is easily computed and can serve as an estimator of Ω. In non-fading or very slow fading cases, [16] showed an optimum lag is n=2N/3. However, in fading situations, the optimum lag is much smaller because for larger n, the fading decorrelates the signal over time and hence the noise has more effect on ϕ_n/n as n gets larger. Fig. 2 illustrates the performance of the estimator ϕ_n/n in both slow and fast fading cases.

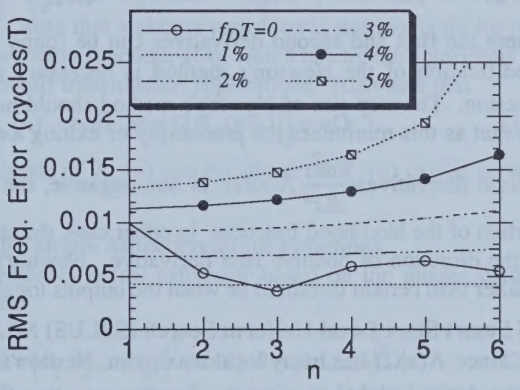


Fig. 2. Performance of the ϕ_n / n estimators. Cases shown are of N=50, $\overline{E}_{eff}/N_0$ =20dB, 100K iterations.

There are two more reasons why only smaller lag autocorrelation terms should be utilized. First, in the **B** matrix, the components near the diagonal are much larger than those off the diagonal in fast fading cases and the larger lag autocorrelation terms contribute less in (10). Second, since the estimator ϕ_n/n has an n-fold ambiguity and the computational burden of phase unwrapping becomes severe with large n. Removing the higher lag terms from (14) gives

$$0 = \sum_{n=1}^{M} n r_n \sin\left(\phi_n + n\Omega\right) \tag{19}$$

where M << N. Further approximation is still needed to solve (19) as this equation has many roots. Since ϕ_n is close to -nΩ for n << N, replacing sin(z) by z in (19) creates a linear estimator of Ω,

$$\hat{\Omega} = -\sum_{n=1}^{M} n r_n \phi_n \Big/ \sum_{n=1}^{M} n^2 r_n . \tag{20}$$

A further simplification of (20) using only ϕ_1 (good in fast fading) and $\phi_3/3$ (good in slow fading) is

$$\hat{\Omega} = -\left(r_1\phi_1 + 3r_3\phi_3\right)/\left(r_1 + 9r_3\right). \tag{21}$$

Notice that all image solutions of Ω (i.e., $-(\phi_n + 2k\pi)/n$ where k is an integer such that $-(\phi_n/2\pi n + k/n)$ is in the candidate range) should be considered in (20) and (21) if the width of the candidate range is larger than 1/3.

3.4 Smart Start Newton Search (SSNS) MLE

Although (20) or (21) provides good performance and low complexity in the estimation of Ω, the performance might still be inadequate for applications requiring tight precision (e.g., large signal constellations). To improve the performance, the estimators of (20) or (21) can serve as the initial starting point for a gradient type search. The well known Newton's method is considered here. The basic recursive formula of the Newton's method is

$$\hat{\Omega}_{k+1} = \hat{\Omega}_k - \frac{d}{d\Omega}\Lambda(\mathbf{x}|\Omega)\Big|_{\Omega=\hat{\Omega}_k} \Big/ \left(\frac{d^2}{d\Omega^2}\Lambda(\mathbf{x}|\Omega)\Big|_{\Omega=\hat{\Omega}_k}\right) \tag{22}$$

where the first and second derivatives can be found in (14) and (16) respectively. Some modification of the Newton's method is necessary since $\Lambda(\mathbf{x}|\Omega)$ is not a pure concave function. The step size of Newton's method should not be allowed to be larger than certain amount as this minimizes the probability of exiting a domain of attraction. Secondly, if the second derivative, $\frac{d^2}{d\Omega^2}\Lambda(\mathbf{x}|\Omega)$, is non-negative, the iteration has reached a non-concave portion of the likelihood function. In either case, the step size is set to be a small step change in the direction of positive first derivative. Newton's method stops when the step size is smaller than certain threshold or when the outputs toggle around some point.

3.5 Smart Start Local Uniform Search (SSLUS) MLE

Since $\Lambda(\mathbf{x}|\Omega)$ has many local maximum, Newton's method frequently does not converge to the desired global maximum. Another way to refine the estimates of (20) or (21) is to search for the point with largest value of $\Lambda(\mathbf{x}|\Omega)$ in the neighborhood of these estimates. This local uniform search increases the computational complexity, but the global maximum of $\Lambda(\mathbf{x}|\Omega)$ is found with high probability and by restricting the search area the complexity is much less than what might be required in a traditional search strategy.

4. Reduced Complexity Frequency Offset Estimator

Often in high speed continuous data communications, the algorithms presented in Section 3 are not computationally feasible and this section examines reduced complexity recursive frequency estimators. From (8), the autocorrelation function of the demodulated pilot symbols given Ω is

$$R_\alpha(n) = E\left[\alpha_k \alpha_{k-n}^* | \Omega\right] = \overline{E}_{eff} J_0(2\pi f_D T n) e^{jn\Omega}. \tag{23}$$

This fact suggests that the phase term of the sample autocorrelation of the demodulated pilot symbols can be used to estimate the Ω. Recently, many algorithms [11-13] have proposed using the uniformly weighted unit lag sample autocorrelation function, $\hat{R}_\alpha(1)$, to estimate Ω. Although the complexity in using $\hat{R}_\alpha(1)$ is low, the performance is not always good enough for mobile communications applications. References [10, 14] combined multiple higher lag sample autocorrelations (i.e., $R_\alpha(n)$ for $n \geq 1$) to provide better performance in estimating Ω. However, these two papers studied only the case of non-fading AWGN channels. For fading channels, algorithms using higher lag sample autocorrelation functions have not been investigated and an algorithm of this type is the major focus of this section.

4.1 Uniformly Weighted Sample Autocorrelation Function

A uniformly weighted sample autocorrelation function has a computationally simple recursion. Define the n-step correlation estimator of the demodulated pilot symbols as

$$\hat{R}_\alpha(n|N) = \frac{1}{N-|n|} \sum_{k=k_{mi}}^{k_{ma}(N)} \alpha_k \alpha_{k-n}^* \tag{24}$$

for any integer n where $k_{mi} = \max(1, 1+n)$ and $k_{ma}(N) = \min(N, N+n)$. The recursion formula for this estimator is

$$\hat{R}_\alpha(n|N+1) = \frac{N-|n|}{N+1-|n|} \hat{R}_\alpha(n|N) + \frac{\alpha_{k_{ma}(N+1)} \alpha_{k_{ma}(N+1)-n}^*}{N+1-|n|}. \tag{25}$$

To simplify notation, we will drop N and use $\hat{R}_\alpha(n)$ for $\hat{R}_\alpha(n|N)$ and k_{ma} for $k_{ma}(N)$ throughout the reminder of this paper. Note that a simple recursion is not available for the nonuniformly weighted sample autocorrelation function, $g(n)$, which is why the algorithms in Section 3 are not as applicable to continuous transmission applications. Also note that

$$E\left[\hat{R}_\alpha(n)|\Omega\right] = R_\alpha(n) \quad and \quad \arg\left\{E\left[\hat{R}_\alpha(n)|\Omega\right]\right\} = n\Omega \tag{26}$$

which means $\hat{R}_\alpha(n)$ is an unbiased estimate of $R_\alpha(n)$ and the phase of $\hat{R}_\alpha(n)$ can be served as a rough estimator of $n\Omega$ if $n<<N$.

4.2 Frequency Offset Estimator Using Sample Autocorrelation Functions

This subsection will develop a frequency offset estimator based on the phases of the sample autocorrelation functions. Define

$$\theta_n = \frac{\arg\left\{\hat{R}_\alpha(n)\right\}}{n}, \quad \boldsymbol{\theta} = \left[\theta_1, \theta_2, ..., \theta_M\right]^T, and \quad \mathbf{1} = [1,1,...,1]^T \tag{27}$$

where $\boldsymbol{\theta}$ and $\mathbf{1}$ are both vectors of length M. Suppose $\boldsymbol{\theta}$ is available to the receiver and M << N. Our goal here is to find the best estimate of Ω given $\boldsymbol{\theta}$. From simulations we found that $\boldsymbol{\theta}$ conditioned on Ω is approximately a jointly Gaussian random vector (also [17] proved the marginal distributions are asymptotic Gaussian). Using this approximation leads to that the optimal estimator of Ω is then a linear function of $\boldsymbol{\theta}$. Note that each component of $\boldsymbol{\theta}$ is an unbiased estimator of Ω . By the joint Gaussian approximation of $\boldsymbol{\theta}$, we have

$$\ln\left\{p(\boldsymbol{\theta}|\Omega)\right\} = f(\boldsymbol{\theta}) - (\boldsymbol{\theta} - \Omega\mathbf{1})^T \mathbf{C}_{\boldsymbol{\theta}}^{-1} (\boldsymbol{\theta} - \Omega\mathbf{1}) \tag{28}$$

where $f(\boldsymbol{\theta})$ is independent of Ω and $\mathbf{C}_{\boldsymbol{\theta}} = E\left[(\boldsymbol{\theta} - \Omega\mathbf{1})(\boldsymbol{\theta} - \Omega\mathbf{1})^T|\Omega\right]$. The standard result of maximum likelihood gives the estimator of Ω as

$$\hat{\Omega} = \mathbf{1}^T \mathbf{C}_{\boldsymbol{\theta}}^{-1} \boldsymbol{\theta} / \mathbf{1}^T \mathbf{C}_{\boldsymbol{\theta}}^{-1} \mathbf{1}. \tag{29}$$

To implement (29) requires calculating $\mathbf{C}_{\boldsymbol{\theta}}$ and an approximate expression is derived in [5]

4.3 Simulation Results

We have assessed the performance of all of the frequency offset estimators presented so far via Monte Carlo simulation and Fig. 3, Fig. 4, and Fig. 5 show the root mean square (RMS) frequency error. In Fig. 3, Fig. 4, and Fig. 5 we have considered fast fading, moderate fading, and stationary-fading respectively. Stationary-fading means f_D =0 (i.e., no vehicle motion) but the MD is still a complex Gaussian random variable (Rayleigh amplitude) and this case is different from the non-fading AWGN model. All the curves correspond to 100K iterations at $\overline{E}_{eff}/N_o$ =20dB. The estimator θ_i is defined by (27). Equation (29) and (21) are two reduced complexity estimator. SSLUS-1 stands for the method proposed in Section 3.5 where the local search range = ±0.01 and search resolution = 0.002. Alternatively SSLUS-2 represents a similar algorithm with a wider local search range (=±0.02) but a coarser search grid (0.003). SSNS denotes the Newton search method in Section 3.4.

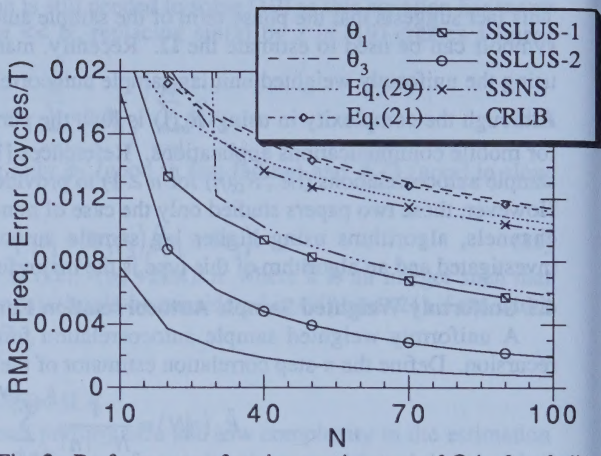


Fig. 3. Performance of various estimators of Ω in fast fading case (f_DT=0.05).

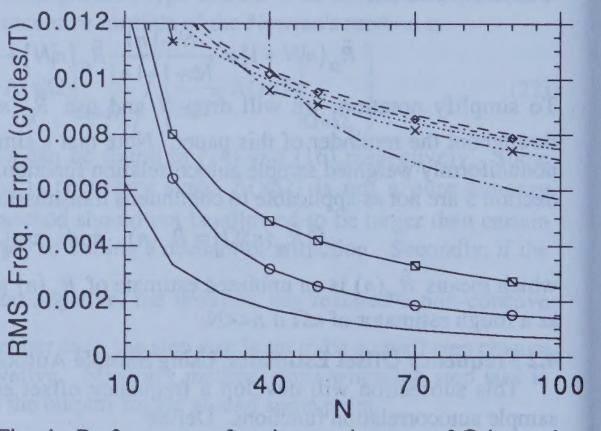


Fig. 4. Performance of various estimators of Ω in moderate fading case (f_DT=0.03).

These curves demonstrate two features. First, a near-optimal frequency offset estimate can be obtained by SSLUS. Gradient search techniques do not work well even for relatively accurate initial guesses due to the multiplicity of local maximums and local non-concavity. The method given in (29) performs well in both slow and fast fading cases. It should be noted that in the stationary-fading case, the weighting coefficients of (20), (21), or (29) are nearly identical to the optimal coefficients derived in [10].

The choice of estimator depends on the detection algorithm used and performance requirement. If a recursive algorithm is employed, (29) is recommended because the uniform weighting in (25) saves a lot of computation or equivalently storage load compared to the renewal procedure of coefficients in (12). However, if the detection is by block processing, the burden of SSLUS becomes more affordable and then SSLUS is a favored choice. Also note that the algorithm in (29) does not work well for small N. If only a small number of

pilot symbols is available or an extreme tight performance is required, SSLUS is recommended.

5. Conclusions

Frequency offset estimation based on transmitted reference signals in Rayleigh flat fading channels has been thoroughly investigated in this paper. The Cramer-Rao performance bound is derived and a near-optimal estimator denoted smart start local uniform search is proposed. For applications requiring a less complex implementation, a frequency offset estimator based on the uniformly weighted sample correlation function of transmitted reference signals and a method of combining the multiple lag correlation terms were proposed and the resulting performance was investigated The associated implementation is simple and high performance is achieved even in very fast fading. Using such frequency estimators can lead to a significant relief of oscillator requirements or a significant increase in the data throughput rate.

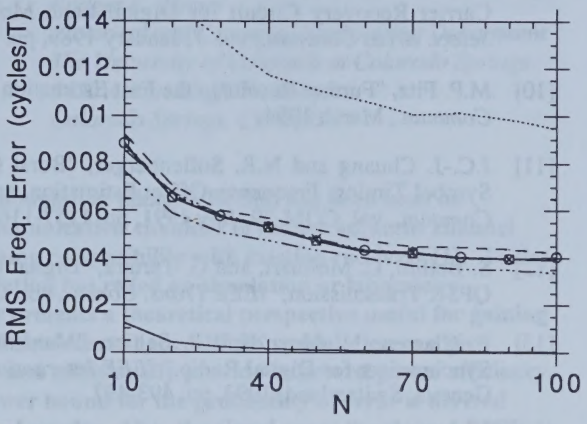


Fig. 5. Performance of various estimators of Ω in zero-fading case (f_DT=0).

References

[1] J.K. Cavers, "An Analysis of Pilot Symbol Assisted Modulation for Rayleigh Faded Channels," *IEEE Trans. Veh. Technol.*, vol. VT-40, November 1991, pp. 686-693.

[2] M.L. Moher and J.H. Lodge, "TCMP - A Modulation and Coding Scheme for Rician Fading Channels," *IEEE J. Select. Areas Commun.*, vol. SAC-7, December 1989, pp. 1347-1355.

[3] M.P. Fitz, "A Dual-Tone Reference Digital Demodulator for Mobile Digital Communications," *IEEE Trans. Veh. Technol.*, vol. VT-42, May 1993, pp. 156-165.

[4] F. Davarian, "Mobile Digital Communication Via Tone Calibration," *IEEE Trans. Veh. Tech.*, vol. VT-36, May 1987, pp. 55-62.

[5] W.-y. Kuo and M.P. Fitz, "Frequency Offset Compensation and Burst Design in Frequency Flat Fading Channel," Technical Report #94-19, Purdue University, May 1994.

[6] F.M. Gardner, "Characteristics of Frequency-Tracking Loops," *IEEE Trans. Commun.*, vol. COM-33, February 1985, pp. 131-139.

[7] F.D. Natali, "AFC Tracking Algorithms," *IEEE Trans. on Comm.*, vol. COM-32, August 1984, pp. 935-947.

[8] T. Alberty and V. Hespelt, "A New Pattern Jitter Free Frequency Error Detector," *IEEE Trans. Commun.*, vol. COM-37, February 1989, pp. 159-163.

[9] S. Saito and H. Suzuki, "Fast Carrier-Tracking Coherent Detection with Dual-Mode Carrier Recovery Circuit for Digital Land Mobile Radio Transmission," *IEEE J. Select. Areas Commun.*, vol. 7, January 1989, pp. 130-139.

[10] M.P. Fitz, "Further Results in the Fast Estimation of a Single Frequency," *IEEE Trans. Commun.*, March 1994,

[11] J.C.-J. Chuang and N.R. Sollenberger, "Burst Coherent Detection with Combined Symbol Timing, Frequency Offset Estimation, and Diversity Selection," *IEEE Trans. Commun.*, vol. COM-39, July 1991, pp. 1157-1164.

[12] S. Bellini, C. Molinari, and G. Tartara, "Digital Frequency Estimation in Burst Mode QPSK Transmission," *IEEE Trans. Comm.*, vol. COM-38, July 1990, pp. 959-961.

[13] F. Classen, H. Meyr, and P. Sehier, "Maximum Likelihood Open Loop Carrier Synchronizer for Digital Radio," *IEEE International Conference on Communications*, Geneva, Switzerland, 1993, pp. 493-497.

[14] M. Luise and R. Reggiannini, "An Efficient Carrier Frequency Recovery Scheme For GSM Receivers," *IEEE Global Telecommunications Conference*, Orlando, FL, 1992, pp. 36-40.

[15] W.C.Y. Lee, *Mobile Communications Engineering*, McGraw-Hill, New York, 1982.

[16] G.W. Lank, I.S. Reed, and G.E. Pollon, "A Semicoherent Detection and Doppler Estimation Statistic," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-9, March 1973, pp. 151-165.

[17] K.S. Miller, *Multidimensional Gaussian Distributions*, John Wiley & Sons Inc., New York, 1964.

Power-Bandwidth Properties of Shaped BPSK

Douglas F. De Boer
Engineering Department
Dordt College
Sioux Center, IA 51250

Rodger E. Ziemer
Electrical and Computer Engineering Department
The University of Colorado at Colorado Springs
1420 Austin Bluffs Parkway
Colorado Springs, CO 80933

Abstract

Shaped biphase-shift keyed modulation (shaped BPSK) has been used on narrowband military satellite communication channels to reduce adjacent channel interference (ACI) while maintaining compatibility with existing receivers [1]. To date most of the analysis of this modulation has relied on simulation or laboratory measurements [2 - 7]. This paper presents a theoretical perspective useful for gaining analytic results. This analysis will show that shaped BPSK modulation can give improved spectral characteristics and additionally provide improved power efficiency, compared to normal BPSK. A lower bound for the probability of error is derived which agrees tightly with simulated results. Also, the signal space for shaped BPSK is found to be three dimensional and typical spectra of shaped BPSK are found by a numerical method.

Introduction

Shaped biphase-shift keyed modulation (shaped BPSK) has been used on narrowband military satellite communication channels to reduce adjacent channel interference (ACI) while maintaining compatibility with existing receivers [1]. To date most of the analysis of this modulation has relied on simulation or laboratory measurements [2 - 7]. This paper presents a theoretical perspective useful for gaining analytic results. This analysis will show that shaped BPSK modulation can give improved power efficiency and spectral characteristics simultaneously, compared to normal BPSK. The analysis was originally motivated by the possibility of decomposing the shaped BPSK modulator into a continuous phase encoder (CPE) and a memoryless modulator (MM) [8]. Such a decomposition for minimum shift keying [9] has shown that continuous phase modulations generally have an implicitly coded structure which can be exploited for error correction. This benefit is in addition to the reduced spectral sidelobes associated with the elimination of abrupt phase transitions. The discussion that follows centers on one particular form of shaped BPSK but the advantages of shaping may apply to a larger set of M-ary phase-shift keyed modulations.

Shaped BPSK modulation is similar to biphase-shift keyed modulation (BPSK) except that when the data calls for a symbol change, a percentage of the symbol interval is used to slew the phase of the carrier π radians while maintaining a constant envelope. The fraction of the symbol interval used for slewing the phase is called the ***Shaping Factor***, δ. Shaping factors of 0.5 are commonly employed to maintain significant compatibility with receivers designed for normal BPSK modulation while still reducing the ACI.

Other methods for reducing the spectral sidelobes of BPSK are possible. For example, the output of a conventional BPSK modulator can be bandpass filtered, but this does not preserve a constant envelope. When filtered BPSK is amplified by a nonlinear device, such as in a satellite, the sidelobes grow back to virtually their unfiltered levels. Shaped modulation

reduces the sidelobes without distorting the envelope. This allows use of nonlinear amplifiers without causing problems of spectral regrowth.

One illustrative method of generating shaped BPSK is to use a linear modulator with both in-phase and quadrature channels. Instead of multiplying a carrier by antipodal rectangular pulse shapes that represent the baseband data, the in-phase channel employs half-cosine shaped antipodal pulses to represent the original binary data, and the quadrature channel, which is not used with ordinary BPSK, employs half-sine shaped pulses to keep the signal envelope constant. (See Figure 1)

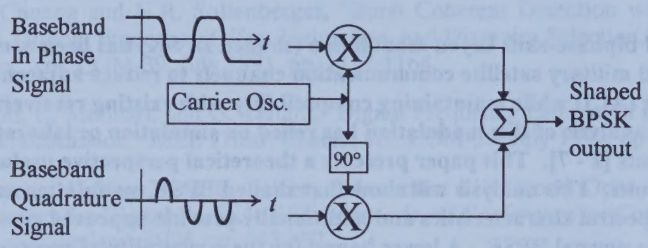


Figure 1. A shaped BPSK modulator

Depending on the polarity of the half-sine pulse in the quadrature channel, the phase can be either advanced or retarded relative to the reference carrier in order to accomplish the desired π radian phase change. In practice a ***vector rotation algorithm*** (VRA) must be adopted when shaped BPSK is employed. A VRA is simply a set of rules used to determine when to advance and when to retard the phase. A popular VRA is to alternate between advancing the phase for two consecutive phase transitions and then retarding the phase for two consecutive phase transitions. This particular VRA satisfies two desirable objectives. First, the average carrier frequency does not shift. This is because the probability of advancing the phase equals the probability of retarding the phase. Second, the average of the baseband signal in the quadrature channel is zero. This is desirable because an average signal in the quadrature channel could result in a phase error in the recovered carrier, given the typical carrier recovery circuits used for BPSK. A non-zero average signal in the quadrature channel also inserts undesirable discrete lines in the spectrum of the signal. This particular VRA (toggle between advancing and retarding the phase with every other data bit transition) is sometimes called Dapper and Hill shaped BPSK (D&H SBPSK) [1] Other VRA's are possible.

A VRA is analogous to a convolutional encoder—it builds memory into the signal. This memory can be exploited by a maximum likelihood sequence estimating detector, such as the Viterbi Algorithm. Alternatively, the transmitter can employ shaped modulation for the reduced ACI while the receiver can be matched to ordinary BPSK. When integrate-and-dump detection matched to ordinary BPSK modulation is used to demodulate a shaped BPSK signal, the detector will suffer at most a $10\log_{10}(1-\delta)$ decibel degradation in power efficiency caused by the shaping, but reduced ACI may be able to more than offset this loss.

Power Spectrum of Shaped BPSK

It is convenient to define some terms related to the complex baseband pulse shapes produced by shaped BPSK. Let:

E_S = Symbol Energy

T = Symbol Period

δ = Shaping Factor ($0 < \delta \leq 1$)

$u(t)$ = Unit Step Function

$r_1(t) = u(t) - u(t - \delta T)$

$r_2(t) = u(t - \delta T) - u(t - T)$

$c(t) = r_1(t)\cos[(\pi t)/(\delta T)]$

$s(t) = r_1(t)\sin[(\pi t)/(\delta T)]$

During any particular symbol interval, the phase of the signal can remain constant at zero, remain constant at π radians, advance from zero to π radians, advance from π radians to 2π radians, retard from 2π radians to π radians, or retard from π radians to zero. This gives six possible complex baseband pulses as follows:

$$s_1(t) = \sqrt{E_s}\left[r_1(t) + r_2(t)\right]$$

$$s_2(t) = -\sqrt{E_s}\left[r_1(t) + r_2(t)\right]$$

$$s_3(t) = \sqrt{E_s}\left[c(t) + \mathrm{j}s(t) - r_2(t)\right]$$

$$s_4(t) = -\sqrt{E_s}\left[c(t) + \mathrm{j}s(t) - r_2(t)\right]$$

$$s_5(t) = \sqrt{E_s}\left[c(t) - \mathrm{j}s(t) - r_2(t)\right]$$

$$s_6(t) = -\sqrt{E_s}\left[c(t) - \mathrm{j}s(t) - r_2(t)\right]$$

Because of the VRA, the next symbol to be transmitted depends on both the next data bit and the present transmitted symbol. This is a Markov process. When calculating the spectrum, this Markov structure must be accounted for. A workable method, although numerically intensive, can be found in [10, p17]. Figure 2 gives a typical comparison of shaped BPSK to normal BPSK.

Three Dimensional Signal Space of Shaped BPSK

The four functions, $r_1(t)$, $r_2(t)$, $c(t)$, and $s(t)$, are useful for calculations with the baseband signals, but just three orthogonal functions could suffice. For example, $\phi_1(t) = r_1(t) + r_2(t)$, $\phi_2(t) = c(t) + (1 - \delta)r_1(t) - \delta r_2(t)$, and $\phi_3(t) = \mathrm{j}s(t)$ form an orthogonal (but not normalized) basis set that could be used for in-phase and quadrature baseband signal processing. The point of interest here is that the symbols themselves are not mutually orthogonal. This means that the metrics used for Viterbi algorithm detection will accumulate correlated noise.

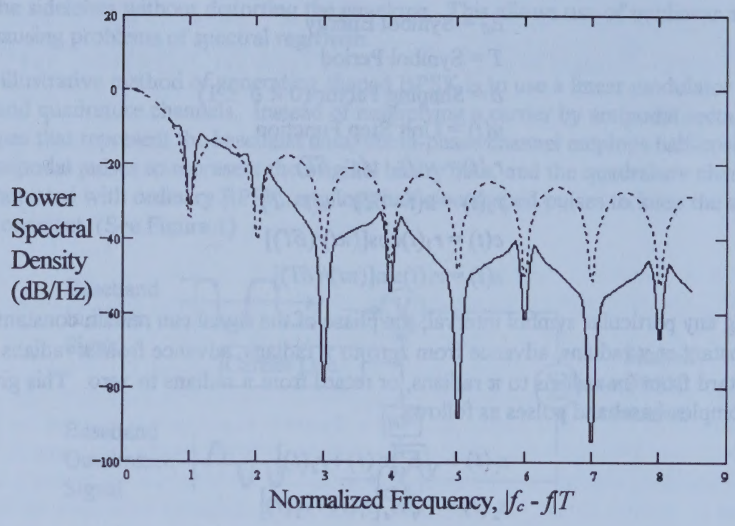


Figure 2, Power spectral density of D&H SBPSK (solid line, δ=0.5) compared to BPSK (dashed line).

Lower Bound on Probability of Bit Error for Viterbi Algorithm Detection

The branch metrics employed in the Viterbi algorithm depend on the chosen VRA. This discussion will assume the VRA of Dapper and Hill mentioned earlier. A state diagram of this VRA is shown in Figure 3. The names of the states refer to the initial phase of the symbol that will be emitted from that state, and where relevant, a "+" sign indicates that the next phase transition will advance the phase, whereas a "-" sign indicates that the next phase transition will retard the phase. All the branch metrics may be derived from four correlation operations carried out on each received symbol. Let $x(t)$ represent the real portion of the received complex baseband signal and let $y(t)$ represent the complex portion. Then the four correlations are:

$$D_1 = \int_0^{\delta T} r_1(t)x(t)dt$$

$$D_2 = \int_{\delta T}^{T} r_2(t)x(t)dt$$

$$D_3 = \int_0^{\delta T} c(t)x(t)dt$$

$$D_4 = \int_0^{\delta T} s(t)y(t)dt$$

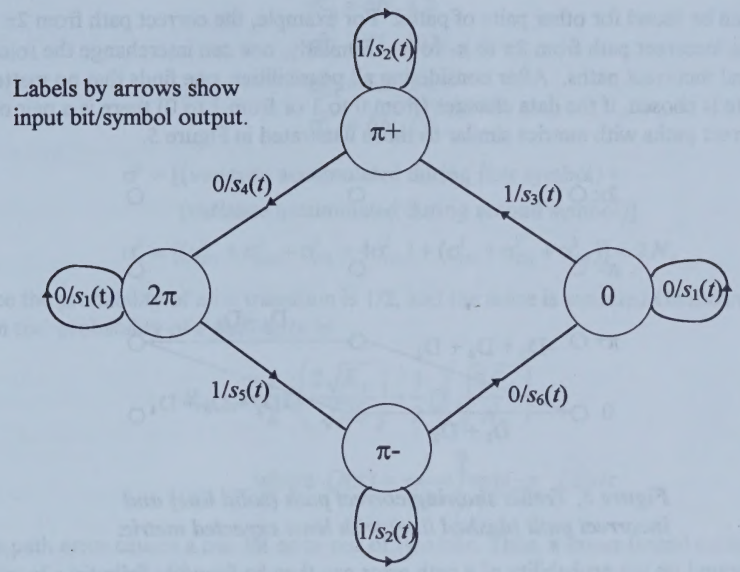


Figure 3, State transition diagram for D&H SBPSK

The branch metrics are then shown on the trellis diagram in Figure 4.

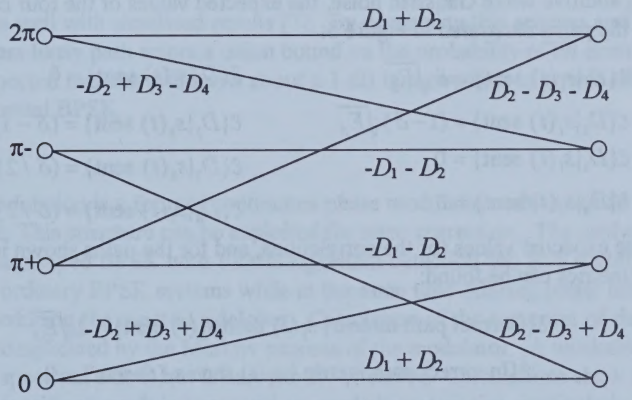


Figure 4, Trellis diagram for D&H SBPSK with VA branch metrics shown.

In order to calculate a lower bound on the probability of a bit error, the shortest incorrect path through the trellis that merges with a correct path must be identified. When the Viterbi algorithm is used to decode convolutional codes, the data chosen to represent the correct path does not matter since the convolutional encoder is linear. In contrast, the state diagram that produces D&H shaped BPSK modulation is nonlinear. It is then necessary to consider different pairs of correct and incorrect paths and find a pair that leads to the least expected difference in the path metrics. The trellis, spanning two symbol intervals, of such a pair of paths is shown in Figure 5. Because of the symmetry of the trellis, identical differences in path

metrics can be found for other pairs of paths. For example, the correct path from 2π to 2π to π- and the incorrect path from 2π to π- to π-. Similarly, one can interchange the roles of the correct and incorrect paths. After considering all possibilities, one finds that no matter which initial state is chosen, if the data changes (from 0 to 1 or from 1 to 0) there is a pair of correct and incorrect paths with metrics similar to those illustrated in Figure 5.

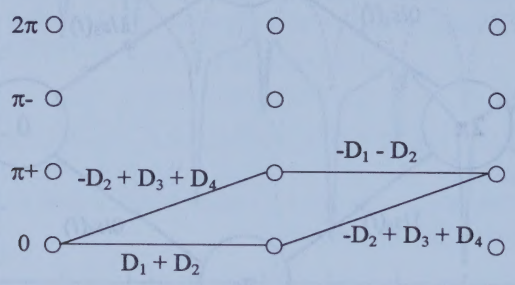


Figure 5, Trellis showing correct path (solid line) and incorrect path (dashed line) with least expected metric.

A lower bound on the probability of a path error can then be found as follows:

$$P_{PE,LB} = \Pr\{\text{Data changes}\} \bullet \Pr\{(\text{Path metric for incorrect path}) - (\text{Path metric for correct path}) > 0\}$$

Assuming additive white Gaussian noise, the expected values of the four correlations are as follows for the paths illustrated in Figure 5:

$$\mathcal{E}\{D_1|s_1(t) \text{ sent}\} = \delta\sqrt{E_s} \qquad \mathcal{E}\{D_1|s_3(t) \text{ sent}\} = 0$$

$$\mathcal{E}\{D_2|s_1(t) \text{ sent}\} = (1-\delta)\sqrt{E_s} \qquad \mathcal{E}\{D_2|s_3(t) \text{ sent}\} = (\delta - 1)\sqrt{E_s}$$

$$\mathcal{E}\{D_3|s_1(t) \text{ sent}\} = 0 \qquad \mathcal{E}\{D_3|s_3(t) \text{ sent}\} = (\delta/2)\sqrt{E_s}$$

$$\mathcal{E}\{D_4|s_1(t) \text{ sent}\} = 0 \qquad \mathcal{E}\{D_4|s_3(t) \text{ sent}\} = (\delta/2)\sqrt{E_s}$$

From these expected values for the correlations, and for the paths shown in Figure 5, the expected path metrics can be found:

$$\mathcal{E}\{\text{Correct path metric} \mid s_1(t) \text{ then } s_3(t) \text{ sent}\} = 2\sqrt{E_s}$$

$$\mathcal{E}\{\text{Incorrect path metric} \mid s_1(t) \text{ then } s_3(t) \text{ sent}\} = 0$$

Since the symbols are not orthogonal, in calculating the noise variance of the path metrics, it is necessary to consider noise correlations. In the first symbol interval shown in Figure 5, both the correct path metric and the incorrect path metric include D_2 in such a way that it reinforces the difference of the path metrics. This means that the variance of D_2 in this instance will contribute a factor of four times the variance of D_2 to the total variance of the difference of the path metrics. Again, in the second symbol interval both path metrics include D_2, but this time with the same sign so that D_2 cancels out of the difference of the path metrics. Below are the variances of the four correlations, assuming the power spectral density of the noise input is N_0 (one sided). These are added, according to the branch metrics just discussed to find the variance of the difference of the two path metrics, σ^2.

$$\sigma_{D1}^2 = \delta N_0 / 2$$
$$\sigma_{D2}^2 = (1-\delta) N_0 / 2$$
$$\sigma_{D3}^2 = \delta N_0 / 4$$
$$\sigma_{D4}^2 = \delta N_0 / 4$$

σ^2 = [(variance accumulated during first symbol) +
(variance accumulated during second symbol)]

$$\sigma^2 = [(\sigma_{D1}^2 + \sigma_{D3}^2 + \sigma_{D4}^2 + 4\sigma_{D2}^2) + (\sigma_{D1}^2 + \sigma_{D3}^2 + \sigma_{D4}^2)] = 2N_0$$

Since the probability of a bit transition is 1/2, and the noise is assumed Gaussian, a lower bound on the probability of a path error is

$$P_{PE,LB} = \frac{1}{2} Q\left(\frac{2\sqrt{E_s}}{\sqrt{\sigma^2}}\right) = \frac{1}{2} Q\left(\sqrt{\frac{2E_s}{N_0}}\right)$$

$$\text{where } Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty \exp(-\tau^2/2) d\tau$$

This path error causes a one bit error out of two bits. Thus, a lower bound on the probability of a bit error, $P_{E,LB}$ is:

$$P_{E,LB} = (1/2) P_{PE,LB} = \frac{1}{4} Q\left(\sqrt{\frac{2E_s}{N_0}}\right)$$

This agrees well with simulated results [7]. By continuing this process and adding the probability of less likely path errors a union bound on the probability of bit error could be found. It is expected that this will show about a 1 dB improvement in power efficiency compared to normal BPSK.

Conclusion

Shaped modulation is a form of continuous phase modulation which has an inherent coded structure. This structure can be exploited for error correction. The analysis in this paper shows that shaped BPSK with Viterbi algorithm detection can give slightly better power efficiency than ordinary BPSK systems while at the same time offering better utilization of bandwidth by reducing the spectral sidelobes. Calculation of the spectrum of shaped modulation is complicated by the Markov process of the modulator. A workable method for calculating the spectrum exists, but is numerically intensive. Calculations show that the smooth phase transitions result in improved spectral characteristics, particularly with regard to adjacent channel interference, compared to BPSK. The benefits of shaping are thought to extend to *M*-ary shaped modulations.

References

[1] M.J. Dapper and T.J. Hill, "SBPSK: A Robust Bandwidth-Efficient Modulation for Hard-Limited Channels," *MILCOM '84 Conf. Record,* Vol. 3, pp. 458-463, Oct. 1984.

[2] R Cofer, E. Franke, O. Johnson, T. Erman, "Shaped PSK in a Digital Modem With Digital Synthesis," *MILCOM '90 Conf. Record, pp 86-92, Oct. 1990.*

[3] M.E. Fox , M.W. Marcellin, "Shaped BPSK and the 5 kHz UHF Satcom Channels," *MILCOM '91 Conf. Record, pp. 15.2.1-15.2.7, Nov. 1991.*

[4] E. Franke, "UHF SATCOM Linear-Phase Constant-Amplitude Waveform Shaping," *MILCOM '92 Conf. Record, pp 1151-1157, Oct 1992.*

[5] R. E. Ziemer, "Adjacent-Channel Interference Effects in Shaped Biphase-Shift Keyed Satellite Communications," *MILCOM '92 Conf. Record*, pp. 837-40, Oct. 1992.

[6] R.E. Ziemer, R. Turcotte, "MLSE of Shaped BPSK in Bandlimited Nonlinear Channels," *MILCOM '93 Conf. Record*, pp. 455-459, Oct. 1993.

[7] R.E. Ziemer, T. Johnson, "Viterbi Algorithm Detection of Shaped BPSK In Adjacent-Channel Interference with Bandlimiting," to appear in *MILCOM '94 Conf. Record.*

[8] B.E. Rimoldi, "A Decomposition Approach to CPM," *IEEE Trans. on Information Theory*, Vol. 32, pp. 260-270.

[9] H. Leib, S. Pasupathy, "Error-Control Properties of Minimum Shift Keying," *IEEE Communications Magazine*, Vol. 31, pp. 52-61, Jan. 1993.

[10] W.C. Lindsey and M.K. Simon, "*Telecommunication Systems Engineering*," Prentice Hall, 1973.

Optimal Data Detection of MPSK in the Presence of Unknown Carrier Phase at Low Complexity

Carl R. Nassar* M. Reza Soleymani†

In many communication systems, transmitters send out signals which carry information in phase, and channels introduce an unknown phase rotation (in addition to additive white Gaussian noise).

A traditional receiver uses a phase locked loop (PLL) to remove the unknown phase. However, in many communication situations, such as communicating over fading channels, communication which involves frequency hopping, and burst communications (such as TDMA), the PLL is inappropriate.

In these cases, one much-used alternative is conventional differential detection, or differential PSK (DPSK). While DPSK resolves the phase acquisition problem in any communication environment, it comes at the cost of an asymptotic 3dB performance degradation.

Recently, researchers have come up with a theoretically optimal way to extend the ideas of differential PSK to blocks of N symbols [1]. They used maximum likelihood (ML) arguments to generate optimal receivers, optimal in the sense of minimizing symbol error rate, given that we have an unknown carrier phase and a block of N symbols. They found a substantial improvement over DPSK by considering only one or two additional symbol intervals of observation. However, one of the drawbacks of their schemes is complexity: complexity increases exponentially with N, making this scheme practical only at block lengths not exceeding 5. This scheme is commonly referred to as multiple symbol differential detection (MSDD).

In our work, we introduce two novel methods for ML data detection in the presence of an unknown phase. Our first method demonstrates an optimal performance and a complexity comparable to that of MSDD. Our second algorithm offers a performance matching the optimal, but has a complexity which increases linearly with N.

In our communication system model, the signal entering the receiver is descibed as

$$r_i = a_i e^{j\theta} + \eta_i \tag{1}$$

Here, a_i represents the complex signal generated at the transmitter. This signal is a differentially encoded MPSK signal, described mathematically as $a_i = e^{j\Phi_i}$ where $\Phi_0 = 0$, $\Phi_i = (\Phi_{i-1} + \phi_i) mod 2\pi, i \geq 1$, and $\phi_i = \{\frac{2\pi l}{M}, l = 0, 1, 2, ..., M-1\}$. θ represents the unknown phase rotation introduced by the channel, and is assumed constant over N symbols. η_i is a complex valued independent Gaussian random variable, with real and

*Department of Electrical Engineering, McGill University, Montreal, Quebec, Canada, H3A 2A7
†SPAR Aerospace Limited, Satellite and Communications Systems Division, Ste-Anne-de-Bellevue.

imaginary parts each having variance $\frac{N_o}{2}$. Equivilantly, we can write

$$\underline{r} = \underline{a}e^{j\theta} + \underline{\eta} \tag{2}$$

where $\underline{r} = (r_i, r_{i+1}, ..., r_{i+N-1})$, and $\underline{a}$ and $\underline{\eta}$ have similar interpretations.

Our goal is to extract the data $\underline{a}$ from the received signal $\underline{r}$. According to the ML detection criteria, the best choice of $\underline{a}$ is

$$\hat{\underline{a}} = arg \max_{\underline{a}} p(\underline{r}|\underline{a}) \tag{3}$$

The posterior distribution $p(\underline{r}|\underline{a})$ is not easily computed, but the augmented posterior $p(\underline{r}|\underline{a}, \theta)$ is well known. Applying the data augmentation algorithm [2] to generate the distribution $p(\underline{r}|\underline{a})$ from $p(\underline{r}|\underline{a}, \theta)$, we find:

$$p(\underline{r}|\underline{a}) = \frac{1}{m} \sum_{j=1}^{m} p(\underline{r}|\underline{a}, \theta^j) \tag{4}$$

where $\{\theta^1, \theta^2, ..., \theta^m\}$ are a sequence of uniformly spaced numbers between $[0, 2\pi]$. This equation can, alternatively, be derived by expressing $p(\underline{r}|\underline{a})$ in integral form as $p(\underline{r}|\underline{a}) = \int p(\underline{r}|\underline{a}, \theta)p(\theta)d\theta$, and then approximating this integral by a summation.

The corresponding ML estimate becomes

$$\hat{\underline{a}} = arg \max_{\underline{a}} \frac{1}{m} \sum_{j=1}^{m} p(\underline{r}|\underline{a}, \theta^j). \tag{5}$$

That is, for each sequence $\underline{a}$, we average $p(\underline{r}|\underline{a}, \theta^j)$ over m equally spaced values of θ^j. The ML sequence $\hat{\underline{a}}$ is simply the $\underline{a}$ with the largest average.

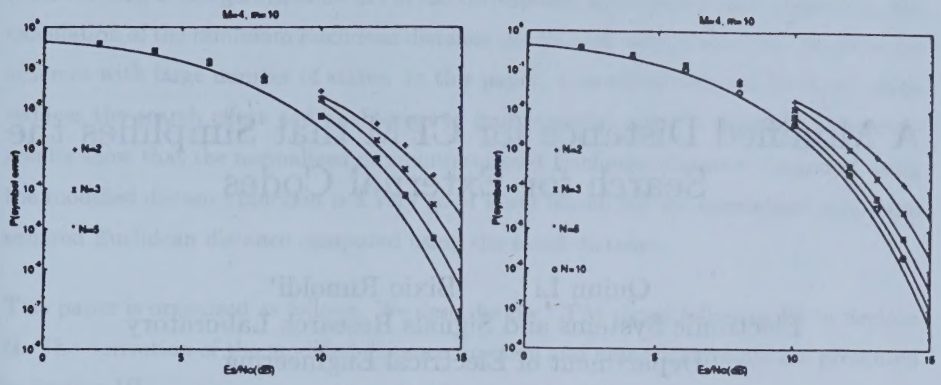
Next, we explore the θ^j's. So far, we have said that $\theta^j, j = \{1, 2, ..., m\}$ should be chosen as equally spaced values over $[0, 2\pi]$. In fact, when we recognize that we are using differential encoding, we see that we could alternatively allow θ^j to be equally spaced values between $[0, \frac{2\pi}{M}]$. Trial-and-error results show that m=10 leads to a performance that is not surpassed by choosing larger values for m.

The performance is demonstrated on the $P(\epsilon)$ vs. $\frac{E_s}{N_o}$ curve of Figure 1(a). Along with each set of simulation points, we have included a solid curve. These curves correspond to the theoretical bounds of MSDD, presented by Divsalar and Simon in [1] (applicable only at large $\frac{E_s}{N_o}$). This figure shows that our scheme achieves the same optimal performance results as MSDD. The bottom curve corresponds to ideal coherent detection of differentially encoded MPSK. As in [1], our performance tends toward this ideal as the block length N increases.

The complexity of our scheme is easy to compute. For each of the M^N sequences, we must compute the m terms $p(\underline{r}|\underline{a}, \theta^j)$. So the complexity of our algorithm is in the order of $m * M^N$, where $m = 10$. The complexity of MSDD is in the order of $N * M^N$. These schemes have more-or-less the same complexity.

Our second algorithm is a low complexity version of the above ML detection scheme. Upon careful examination of equation (5), it becomes clear that an alternative is

$$\hat{\underline{a}} = arg \max_{\underline{a}} (arg \max_{\theta^j} p(\underline{r}|\underline{a}, \theta^j)) \tag{6}$$



(a) (b)

Figure 1: $P(\epsilon)$ vs. $\frac{E_s}{N_o}$ curves for: (a) the averaging method with M=4, m=10, N=2,3,5; (b) the limited search algorithm with M=4, m=10, N=2,3,5,10. (M=2,8,16 similar.)

where $\theta^j \epsilon \{\theta^1, ..., \theta^m\}$. This is because the m term sum $\sum_{j=1}^{m} p(\underline{r}|\underline{a}, \theta^j)$ is dominated by the largest term of the sum, $arg \max_{\theta^j} p(\underline{r}|\underline{a}, \theta^j))$. This equation can be reexpressed as

$$\hat{\underline{a}} = arg \max_{\theta^j} (arg \max_{a_i} p(r_i|a_i, \theta^j) \times ... \times arg \max_{a_{i+N-1}} p(r_{i+N-1}|a_{i+N-1}, \theta^j)) \qquad (7)$$

This can be carried out as follows. First, hold θ^j at θ^1 and compute $arg \max_{\underline{a}} p(\underline{r}|\underline{a}, \theta^j)$ according to: choose the a_i which maximizes $p(r_i|a_i, \theta^j)$, the a_{i+1} which maximizes $p(r_{i+1}|a_{i+1}, \theta^j)$, and so on. Repeat this with θ^j held fixed at θ^2,θ^3, up to θ^m. This results in m sequences, one per θ^j. Choose the one of these m sequences which has the largest product $p(r_i|a_i, \theta^j) \times p(r_{i+1}|a_{i+1}, \theta^j) \times ... \times p(r_{i+N-1}|a_{i+N-1}, \theta^j)$.

Simply put, this algorithm is an attempt to compute the joint ML estimate of $\underline{a}$ and θ using an "exhaustive" search, where rather than searching over the infinite values of θ, we search only over m of these.

This algorithm requires computing N*M*m values of $p(r_{i+k}|a_{i+k}, \theta^j)$, and so its complexity increases only linearly with N rather than exponentially. Using a theoretical presentation, we can show that, as $N \rightarrow \infty$, the performance of this scheme matches ideal coherent detection of MPSK with differential encoding. Figure 1(b) details our simulated $P(\epsilon)$ vs. $\frac{E_s}{N_o}$ performance curves. Here again, our results match the theoretical curves of the optimal MSDD.

This time we see that we have achieved the same performance as the optimal MSDD presented in [1], while requiring a complexity increasing only linearly with block size N.

References

[1] D. Divsalar and M.K. Simon, "Multiple-symbol differential detection of MPSK," *IEEE Trans. Commun.*, Vol. 38, pp.300-308, Mar.1990.

[2] M.A. Tanner and W.H. Wong, "The calculation of posterior distributions by data augmentation (with discussion)," *Journal of the American Statistical Association*, vol. 82, pp.528-550, June 1987.

A Modified Distance for CPM that Simplifies the Search for External Codes

Quinn Li Bixio Rimoldi*
Electronic Systems and Signals Research Laboratory
Department of Electrical Engineering
Washington University
St. Louis, MO 63130 USA

September 28, 1994

Abstract

A modified Euclidean distance function for Continuous Phase Modulation (CPM) that substantially simplifies the search for good external convolutional codes has been derived. Numerical results show that the normalized minimum squared Euclidean distance computed using the modified distance function is a very good lower bound for the normalized minimum squared Euclidean distance computed using the exact distance.

I Introduction

Continuous Phase Modulation (CPM) schemes are a class of constant-envelope modulation schemes. The constant-envelope nature of the signals makes these schemes suitable for transmission over nonlinear and/or fading channels, such as the satellite and the mobile radio channels. Moreover, CPM signals have good spectral properties due to their phase continuity.

Much work on coding for CPM has been done to improve its energy efficiency [1, 2, 3, 4, 5, 6]. Since the Euclidean distance between CPM signals is not a function of the difference

*Supported by National Science Foundation grant NCR-9109944, NCR-9357689.

(over the ring of integers modulo M) of the corresponding encoder output sequences, the calculation of the minimum Euclidean distance can be very computationally intensive for schemes with large number of states. In this paper, a modified distance function which reduces the search effort and enables us to find powerful codes is derived. Numerical results show that the normalized minimum squared Euclidean distance computed using the modified distance function is a very good lower bound for the normalized minimum squared Euclidean distance computed using the exact distance.

This paper is organized as follows. We describe the CPM signal following [7] in Section II. The derivation of the modified distance function and numerical results are presented in Section III.

II Decomposed Model of CPM

The transmitted signal of CPM can be described as (see [3, 7]):

$$s(t, \boldsymbol{U}) = \sqrt{2E_s/T_s}\cos(2\pi f_o t + \psi(t, \boldsymbol{U}) + \psi_o), \ t \geq 0, \tag{1}$$

where

$$\psi(t, \boldsymbol{U}) = 4\pi h \sum_{i=0}^{\infty} U_i f(t - iT_s), \ t \geq 0, \tag{2}$$

is the *information-carrying phase*, E_s is the *symbol energy*, T_s is the *symbol time*, f_o is the *carrier frequency*, and ψ_o is an initial phase shift which can be assumed to be zero without loss of generality. The parameter h is referred to as the *modulation index*, and is defined by

$$h = K/P,$$

where K and P are relatively prime positive integers. The *input sequence* $\boldsymbol{U}$ is simply an M-ary sequence with $U_i \in \{0, 1, \cdots, M-1\}$, and the *phase response* $f(t)$ satisfies

$$f(t) = \begin{cases} 0, & t \leq 0 \\ \frac{1}{2}, & t > LT_s, \end{cases} \tag{3}$$

where L is a positive integer called the *memory* of the system, and $f(t)$ is continuous and monotonically increasing in the interval $(0, LT_s]$.

It is straightforward to verify that the transmitted signal does not change if we substitute

$\psi'(t, \boldsymbol{U})$ for $\psi(t, \boldsymbol{U})$ in (1), where

$$\psi'(t, \boldsymbol{U}) = 2\pi h R_P[\sum_{i=0}^{n-L} U_i] + 4\pi h \sum_{i=n-L+1}^{n} U_i f(t - iT_s), \quad nT_s \leq t < nT_s + T_s, \tag{4}$$

and $R_P[.]$ denotes the remainder after division of the enclosed number by P. Hence, $U_n, U_{n-1}, \cdots, U_{n-L+1}$ and $R_P[\sum_{i=0}^{n-L} U_i]$ completely specify the transmitted signal in the n-th interval, and a CPM signal can be generated as shown in Figure 1 [7].

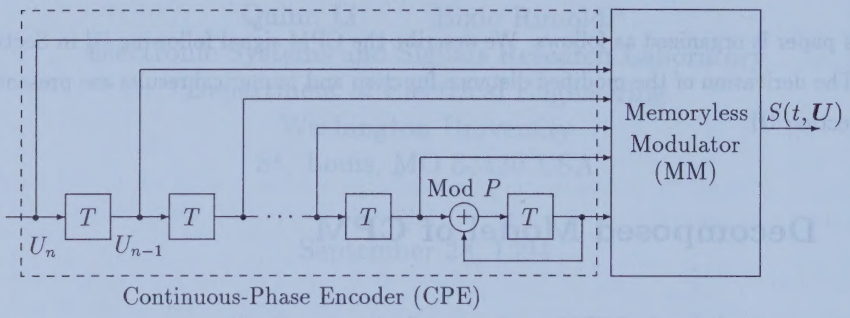


Figure 1: Block diagram of an M-ary CPM scheme with modulation index $h = K/P$. The CPM scheme is decomposed into a CPE followed by a MM.

III A Modified Squared Euclidean Distance

We assume the additive white Gaussian channel model. The objective of the external encoder is to maximize the *normalized minimum squared Euclidean distance* of the encoded scheme,

$$d^2_{min} = \frac{1}{2E_b} \min_{\boldsymbol{U} \neq \boldsymbol{U}'} D^2(\boldsymbol{U}, \boldsymbol{U}'), \tag{5}$$

where $E_b = \frac{E_s}{r \log_2 M}$ is the energy per information bit and r is the rate of the external encoder. The *squared Euclidean distance* (SED) $D^2(\boldsymbol{U}, \boldsymbol{U}')$ between two signals $s(t, \boldsymbol{U})$ and $s(t, \boldsymbol{U}')$ can be written as

$$D^2(\boldsymbol{U}, \boldsymbol{U}') = \sum_{n=0}^{\infty} D_n^2(\boldsymbol{U}, \boldsymbol{U}'), \tag{6}$$

where $D_n^2(\boldsymbol{U}, \boldsymbol{U}')$ is the *incremental squared Euclidean distance* (ISED) in the n-th interval given by

$$D_n^2(\boldsymbol{U}, \boldsymbol{U}') = \int_{nT_s}^{nT_s+T_s} [s(t, \boldsymbol{U}) - s(t, \boldsymbol{U}')]^2 dt$$

$$= 2E_s[1 - \frac{1}{T_s}\int_{nT_s}^{nT_s+T_s} \cos(\Delta\psi(t,\boldsymbol{\varepsilon}))dt], \quad (7)$$

where $\boldsymbol{\varepsilon}$ is the difference (over the reals) of the input sequences, i.e., $\boldsymbol{\varepsilon} := \boldsymbol{U} - \boldsymbol{U}'$. The phase difference $\Delta\psi(t,\boldsymbol{\varepsilon})$ can be written using (4) as

$$\begin{aligned}\Delta\psi(t,\boldsymbol{\varepsilon}) &= \psi(t,\boldsymbol{U}) - \psi(t,\boldsymbol{U}') \\ &= 2\pi h R_P[\sum_{i=0}^{n-L}\varepsilon_i] + 4\pi h \sum_{i=n-L+1}^{n}\varepsilon_i f(t-iT_s),\ nT_s \le t < nT_s + T_s. \quad (8)\end{aligned}$$

Notice that $D_n^2(\boldsymbol{U},\boldsymbol{U}')$ depends on $\boldsymbol{\varepsilon}$ via the $(L+1)$-tuple $(\varepsilon_n, \varepsilon_{n-1}, \cdots, \varepsilon_{n-L+1}, R_P[\sum_{i=0}^{n-L}\varepsilon_i])$.

From (6)-(8), we can see that $D^2(\boldsymbol{U},\boldsymbol{U}')$ does not depend only on $R_M[\boldsymbol{U}-\boldsymbol{U}']$. Hence, we can not simplify the computation of d_{min}^2 by assuming that $\boldsymbol{U}$ is the zero sequence, denoted by $\mathbf{o}$. In this section, we derive a new distance function $\tilde{D}^2(\boldsymbol{U},\boldsymbol{U}')$ called *modified* SED, which has the following two properties:

1. $\tilde{D}^2(\boldsymbol{U},\boldsymbol{U}')$ depends only on $R_M[\boldsymbol{U}-\boldsymbol{U}']$, i.e., $\tilde{D}^2(\boldsymbol{U},\boldsymbol{U}') = \tilde{D}^2(R_M[\boldsymbol{U}-\boldsymbol{U}'],\mathbf{o})$;
2. $\tilde{D}^2(\boldsymbol{U},\boldsymbol{U}') \le D^2(\boldsymbol{U},\boldsymbol{U}')$.

As a result, if we use $\tilde{D}^2(\boldsymbol{U},\boldsymbol{U}')$ instead of the correct measure $D^2(\boldsymbol{U},\boldsymbol{U}')$, we can assume $\boldsymbol{U} = \mathbf{o}$ and it will be guaranteed that the *modified normalized minimum squared Euclidean distance*, $\tilde{d}_{min}^2$ (see (12)), is a lower bound for d_{min}^2.

Let $\mathcal{U}$ be the set of input sequences,

$$\mathcal{U} = \{U_0, U_1, \cdots : U_i \in \{0,1,\cdots M-1\}\}.$$

Then $\Delta\boldsymbol{U} := R_M[\boldsymbol{U}-\boldsymbol{U}'] \in \mathcal{U}$ for any $\boldsymbol{U},\boldsymbol{U}' \in \mathcal{U}$. Define the *modified* SED associated with $\Delta\boldsymbol{U}$ as

$$\tilde{D}^2(\boldsymbol{U},\boldsymbol{U}') = \tilde{D}^2(\Delta\boldsymbol{U},\mathbf{o}) := \sum_{n=0}^{\infty}\tilde{D}_n^2(\Delta\boldsymbol{U},\mathbf{o}), \quad (9)$$

where $\tilde{D}_n^2(\Delta\boldsymbol{U},\mathbf{o})$ is the *modified* ISED associated with $\Delta\boldsymbol{U}$ in the n-th interval, and is defined by

$$\begin{aligned}\tilde{D}_n^2(\Delta\boldsymbol{U},\mathbf{o}) &:= \min_{\boldsymbol{U},\boldsymbol{U}':R_M[\boldsymbol{U}-\boldsymbol{U}']=\Delta\boldsymbol{U}} D_n^2(\boldsymbol{U},\boldsymbol{U}') \quad (10) \\ &= \min_{\varepsilon_i\in\xi(\Delta U_i),\, i=n-L+1,\cdots,n} D_n^2(\boldsymbol{U},\boldsymbol{U}'), \quad (11)\end{aligned}$$

where $\xi(\Delta U_i) := \{\Delta U_i, \Delta U_i - M\}$ and $D_n^2(\boldsymbol{U}, \boldsymbol{U}')$ is given in (7). The step from (10) to (11) follows from the fact that $D_n^2(\boldsymbol{U}, \boldsymbol{U}')$ depends on $\boldsymbol{U}, \boldsymbol{U}'$ only via ε_i for $i = n - L + 1, \cdots, n$ and $R_P[\sum_{i=0}^{n-L} \varepsilon_i]$ (see (8)), and the fact that $P = M$ implies

$$R_P[\sum_{i=0}^{n-L} \varepsilon_i] = R_M[\sum_{i=0}^{n-L} \varepsilon_i] = R_M[\sum_{i=0}^{n-L} \Delta U_i].$$

Now the *modified normalized minimum squared Euclidean distance*, $\tilde{d}_{min}^2$, is obtained from the minimum of $\tilde{D}^2(\Delta \boldsymbol{U}, \mathbf{o})$ over all $\Delta \boldsymbol{U} \neq \mathbf{o}$, i.e.,

$$\tilde{d}_{min}^2 := \frac{1}{2E_b} \min_{\boldsymbol{U}, \boldsymbol{U}': R_M[\boldsymbol{U} - \boldsymbol{U}'] \neq \mathbf{o}} \tilde{D}^2(\boldsymbol{U}, \boldsymbol{U}') = \frac{1}{2E_b} \min_{\Delta \boldsymbol{U} \neq \mathbf{o}} \tilde{D}^2(\Delta \boldsymbol{U}, \mathbf{o}). \tag{12}$$

Clearly, $\tilde{d}_{min}^2 \leq d_{min}^2$.

How tight is the lower bound? Both $\tilde{d}_{min}^2$ and d_{min}^2 are computed for ternary and quaternary coded LREC schemes with $L = 1, 2$, and 3, and the results are listed in Table I and II. We can see from these results that for most cases $\tilde{d}_{min}^2$ equals d_{min}^2. For those cases where $\tilde{d}_{min}^2$ does not equal d_{min}^2, $\tilde{d}_{min}^2$ is a very good lower bound on d_{min}^2.

TABLE I – $\tilde{d}_{min}^2$ and d_{min}^2 FOR CODED TERNARY LREC SCHEMES WITH $h = 1/3$ USING RATE 1/2 TERNARY RING CONVOLUTIONAL ENCODERS

L	ν	$\tilde{d}_{min}^2$	d_{min}^2
1	1	4.237	4.237
	2	5.891	6.096
	3	7.284	7.284
2	1	3.200	3.200
	2	4.199	4.321
3	1	2.096	3.970

TABLE II – $\tilde{d}_{min}^2$ and d_{min}^2 FOR CODED QUATERNARY LREC SCHEMES WITH $h = 1/4$ USING RATE 1/2 QUATERNARY RING CONVOLUTIONAL ENCODERS

L	ν	$\tilde{d}_{min}^2$	d_{min}^2
1	1	4.302	4.302
	2	6.241	6.241
2	1	3.016	3.016
	2	3.727	3.727

Hence, we can first use the modified distance function $\tilde{D}^2(.)$ to search for codes with large number of states, and once good codes (codes with high $\tilde{d}_{min}^2$) are found, the exact

d^2_{min} can be computed. It should be pointed out that for some cases, the optimal codes found by using $\tilde{D}^2(.)$ are different than those found using $D^2(.)$. Hence, the best codes found by using $\tilde{D}^2(.)$ may not be optimal in the usual sense. Nevertheless, our modified distance measure makes it feasible to find good codes for encoded systems with large number of states.

IV Conclusion

A modified Euclidean distance function for CPM that substantially simplifies the search for good external convolutional codes has been derived. Numerical results show that the normalized minimum squared Euclidean distance computed using the modified distance function is a very good lower bound for the exact normalized minimum squared Euclidean distance.

References

[1] G. Lindell, C.-E. W. Sundberg, and T. Aulin, "Minimum Euclidean distance for the best combination of short rate 1/2 convolutional codes and CPFSK modulation," *IEEE Transactions on Information Theory*, vol. 30, pp. 509–520, May 1984.

[2] S. V. Pizzi and S. G. Wilson, "Convolutional coding combined with Continuous Phase Modulation," *IEEE Transactions on Communications*, vol. 33, pp. 20–29, January 1985.

[3] J. B. Anderson, T. Aulin, and C. E. Sundberg, *Digital Phase Modulation.* New York: Plenum Press, 1986.

[4] P. K. M. Ho and P. J. McLane, "Spectrum, distance, and receiver complexity of encoded Continuous Phase Modulation," *IEEE Transactions on Information Theory*, vol. 34, pp. 1021–1032, September 1988.

[5] B. Rimoldi, "Design of coded modulation systems for bandwidth and energy efficiency," *IEEE Transactions on Communications*, vol. 37, pp. 897–905, September 1989.

[6] K. Balachandran, J. B. Anderson, W. A. Finamore, and C. J.-L. Pimentel, "Better rate 1/2 coded Continuous Phase Modulation schemes and the M-algorithm," *IEEE Transactions on Information Theory*, vol. 39, pp. 1687–1694, September 1993.

[7] B. Rimoldi, "A decomposition approach to CPM," *IEEE Transactions on Information Theory*, vol. 34, pp. 260–270, March 1988.

AN UPPER BOUND ON SYNCHRONIZATION PROBABILITY FOR MPPM SEQUENCES

Ramaiah Velidi and Costas N. Georghiades
Electrical Engineering Department
Texas A&M University
College Station, Texas 77843-3128

I. Introduction

The performance of a symbol synchronizer operating on finite-length, randomly generated, multi-pulse pulse-position modulation (MPPM) sequences is severely limited by those sequences. We derive an upper bound on the probability of correct symbol synchronization, P_{cs}, for MPPM sequences, when slot boundaries are known.

MPPM has recently received attention in the domain of high speed intersatellite communications, primarily because it has a capacity advantage and significantly reduces the required transmission bandwidth compared to PPM [1]. In MPPM, optical pulses are transmitted in p out of the Q slots in a symbol interval, referred to as (Q,p) MPPM. There are $M = \binom{Q}{p}$ MPPM symbols for a given Q and p. Given a sequence of $(N+1)$ MPPM symbols and an observation window of NQ slots, we consider the problem of locating the MPPM symbol boundaries. Any sequence of consecutive NQ slots that is observed by sliding the observation window to the right by $m \in \{1,2,\ldots,(Q-1)\}$ slots is a *subsequence*. If each consecutive Q slots in that window is a (Q,p) MPPM symbol, then the subsequence is a *valid subsequence*. The performance of the symbol synchronizer is limited by the nature of the randomly transmitted MPPM sequence, because there could be more than one valid MPPM subsequence.

II. The Bound

Toward deriving a bound on the random sequence-limited probability, P_{rsl}, we state without proof some claims (*only sufficient and not necessary*) regarding the behavior of random MPPM subsequences.

Claim 1: An NQ slot subsequence in an observation window which is shifted from its original position (0^{th} *slot*) to the right by $m < p$ slots, will be a valid subsequence if each symbol in the $(N+1)$ symbol sequence has pulsed slots in the $m^{th}, (m+1)^{st}, \ldots, (Q-1)^{st}$ slot positions.

Claim 2: An NQ slot subsequence in an observation window which is shifted from its original position to the right by $p \leq m \leq (Q-p)$ slots will be a valid subsequence if each symbol that appears in the $(N+1)$ symbol sequence has its $m^{th}, (m+1), \ldots, (Q-1)^{st}$ *or* $0^{th}, 1^{st}, \ldots, (Q-m)^{th}$ slots pulsed.

Claim 3: An NQ slot subsequence in an observation window which is shifted from its original position to the right by $m > (Q-p)$ slots will be a valid subsequence if all the MPPM symbols in the $(N+1)$ symbol sequence have pulsed slots in the $0^{th}, 1^{st}, \ldots, (m-1)^{st}$ slot positions.

Let D_k be the number of $(N+1)$ MPPM symbol sequences, which contain exactly k valid subsequences, $\forall k \in \{0, 1, \ldots, (Q-1)\}$. Using the above claims we can show that

$$D_k \leq (k+1)\left[T_1(k)^{N+1} - 2T_2(k)^{N+1} + T_3(k)^{N+1}\right], \tag{1}$$

$$T_i(k) = \begin{cases} \binom{Q-\alpha k-i+1}{p}, & Q-\alpha k-i+1 \geq p \\ 0, & \text{otherwise.} \end{cases}, \tag{2}$$

where, $i = 1, 2, 3$, $\alpha = 1$ for $i = 1, 2$ and $\alpha = 2$ for $i = 3$. The random sequence limited probability is given by

$$\mathsf{P}_{rsl} = \frac{1}{M^{N+1}} \sum_{k=0}^{Q-1} \frac{D_k}{k+1}$$

where, D_k is defined in Eq. (1). Therefore, performing the required summation,

$$\mathsf{P}_{cs} \leq \mathsf{P}_{rsl} \leq 1 - (1 - p/Q)^{N+1}. \tag{3}$$

Sequences that have no valid subsequences are referred to as *synchronizable.* A brief search shows that two synchronizable pairs (two MPPM symbols) have back-to-back pulses. But, there exist other synchronizable sequences as well: if the number of back-to-back non-pulsed slots is greater than $(Q-p)$ and the first and second symbol in the pair have their 0^{th} and $(Q-1)^{st}$ slots pulsed, respectively, *or* the number of back-to-back pulsed slots is greater than p and the first and second symbol in the pair have their 0^{th} and $(Q-1)^{st}$ slots non-pulsed, respectively, then the pair will be synchronizable. A finite length MPPM sequence is synchronizable if and only if it contains one or more synchronizable pairs. As suggested in [2] for PPM, the performance ceiling predicted by Eq. (1) for random sequences can be removed by periodically inserting a synchronizable pair in the data stream. Designing synchronizable codes [3] for the optical MPPM channel, to improve the performance ceiling, would be an interesting future project.

References

[1] H.Sugiyama and K.Nosu, "MPPM : A method for improving the band-utilization efficiency in optical PPM", *J. Lightwave Technol.*, vol.-7, pp. 465-472, Mar. 1989.

[2] C.N.Georghiades, "On the Synchronizability and Detectability of Random PPM Sequences", *IEEE Trans. on Info. Theory*, No. 1, pp. 146-156, Jan. 1989.

[3] A.R.Calderbank and C.N. Georghiades, "Synchronizable Codes for the Optical OPPM Channel", To appear, *IEEE Trans. on Info. Theory*, July 1994.

Optimal Optical-Domain Filtering for Fiber-Optic Communications

Maïté Brandt-Pearce
Department of Electrical Engineering
University of Virginia
Charlottesville, VA 22903

ABSTRACT

A method of designing fiber-optic filters to minimize the error probability in optical communications is proposed. These filters can be implemented via fiber-optic tapped-delay lines. If a photodetector with infinite bandwidth impulse response is used, the design method relies on optimizing the error probability over all filter impulse responses. The optimal filter is shown to be a 0-1 filter under some conditions. If a finite bandwidth photodetector is used, the optimization is over the Chernoff bound. The resulting optimal filter is similar but with sloped transitions. This filter design method is applied to a communication system suffering from intersymbol interference. The optimal filter is found to significantly improve the performance of the system.

1 Introduction

A general method of designing optical linear filters to minimize the probability of bit error in an optical communication system is presented. These filters can be designed to equalize a dispersion-limited or colored noise channel, as well as to pulse shape or reject multiple-access interference [1]. The advantage of filtering in the optical domain, before photodetection, is the availability of the full optical bandwidth for signal processing and the reduction in noise. Quantum noise in optical communications is proportional to the signal energy received. Irrelevant or "nuisance" energy can be removed from the incident optical signal and therefore reduce noise. Broadband FIR and IIR optical filters can be implemented easily using fiber tapped delay lines or integrated optics [2, 3, 4]. The primary restriction on the impulse response of the filter is nonnegativity. It has been shown that any nonnegative filter response can be approximated using fiber-optic filters [2].

The only type of electronic processing considered here is a simple threshold detector. The eventual goal is to perform all processing optically, either through a linear filter, adaptive equalizer, or nonlinear processing. The work presented here is a first step is realizing this goal.

Traditional optical filters have been designed using a maximum signal-to-noise ratio or a minimum-mean-square error as the optimality criteria [3, 5]. As long as the received energies are high, these methods yield good results, since Gaussian probability distribution approximations hold. However, when the received energies are low, the Poisson noise dominates and filters should be designed accordingly. In this paper we derive an optimal filter using a Poisson model for the received signal. The only other noise source is the constant dark current present at the

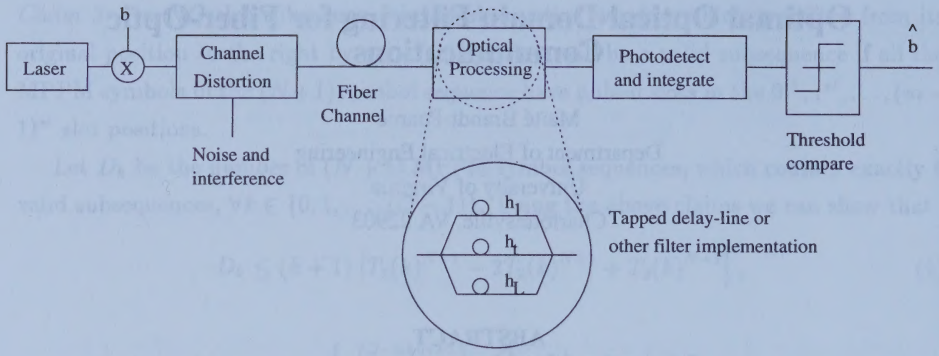


Figure 1: Block diagram of fiber-optic receiver for binary communications.

photodetector. A restriction on the impulse response in addition to nonnegativity is a less than unit amplitude for all time.

The minimum probability of error filter is obtained by calculating the variation of the error probability or the variation on a bound to the error probability. In the case of Poisson observation and no photodetection filtering, a closed-form expression for the error probability can be obtained and it is used to derive the optimal filter. If it is desirable to consider the photodetector as having a bandlimited impulse response, a closed-form approximation based on the Chernoff bound is available and can be used to define the best filter. This approximation is asymptotically tight as the energy increases.

The problem is described in Section 2. The optimal filter for different cases is found in Section 3. As an illustration of the method, the optimal filter is found in Section 4 for a dispersive channel under different photodetector response assumptions.

2 Problem Formulation

Consider the binary optical communication system illustrated in Figure 1. The system is composed of an optical source being on-off modulated by the information symbols. The resulting beam travels down a fiber where it is subjected to channel distortion in the form of dispersion or additive noise. A semi-classical model of optics is adopted: the optical signal is modeled as a continuous time signal, while the output of the photodetector is a point process. The problem can be formulated as a hypothesis testing problem where one of two optical signals is received, $s_0(t)$ or $s_1(t)$, depending on whether the symbol sent $b = 0$ or 1. The observation is a doubly stochastic Poisson point process with rate function proportional to the light intensity incident on the photosensitive area, plus a dark current term. Direct detection is assumed, and the received signal is assumed to have a short coherence time, so that no coherent combining occurs in filtering. The received signal is filtered optically by a filter with impulse response $h(t)$. The rate function on the Poisson process under hypothesis i becomes

$$\lambda_i(t) = \int_{-\infty}^{t} h(t-\tau)s_i(\tau)\,d\tau + \lambda_d, \tag{1}$$

where λ_d is the dark current contribution to the received intensity. We will assume that the information signal $s_i(t)$ is a stochastic process instead of a known function. The types of system distortions that can be accommodated by this model include additive noise channels, dispersive channels, multiple access channels, etcetera.

Assume that the observation is restricted to signal at the output of the photodetector accumulated over $(0, T]$, yielding an observation variable X. No further electronic processing is done on the variable X before thresholding. Two cases are considered for the photodetection process: an ideal and a nonideal photodetector response. In the ideal case, the photodiode has infinitely fast response and the observation is Poisson. Let

$$\Lambda_i = \int_0^T \lambda_i(t)\, dt + T\lambda_d \tag{2}$$

be the parameter of the Poisson random variable X under hypothesis $i = 0$ or 1 in this case. We can simplify the notation by simply letting

$$\Lambda_i = \int_{-\infty}^T h(t-\tau)s_i(\tau)\, d\tau + T\lambda_d. \tag{3}$$

In the case on a nonideal optical-to-electrical conversion, the photodetector has an impulse response $g(t)$ assumed known. This is a better model than the ideal one when the bandwidth of the device is smaller than the bandwidth of the information signal. The observation in this case is an accumulation of a filtered Poisson process with intensity function λ_i.

3 Filter Design

In this section we define a method for designing optimal filter for the system described above. In the case of an ideal photodetector, the optimal filter is found by optimizing the error probability. This is the first problem addressed. The optimal filter for a system with a nonideal photodetector is then derived by minimizing the Chernoff bound to the error probability.

3.1 Ideal Photodetection

Assuming an ideal photodetector, we have a decision statistic equal to our observation X to compare to a threshold γ. The probability of error which is our cost function can be written as

$$P_e = E\left[\sum_{n=0}^{\lfloor\gamma\rfloor} \frac{1}{n!} e^{-\Lambda_1}(\Lambda_1)^n + \sum_{n=\lceil\gamma\rceil}^{\infty} \frac{1}{n!} e^{-\Lambda_0}(\Lambda_0)^n\right]. \tag{4}$$

The expectation $E[\cdot]$ is over the random parameters $\Lambda_{(i)}$. The constraint on $h(t)$ is a pointwise inequality constraint

$$0 \le h(t) \le 1 \quad \forall t. \tag{5}$$

It should be noted that such constraints are equivalent to an infinite number of constraints since they affect every t individually, and can therefore not be included by a Lagrange multiplier approach. We will first consider the unconstrained problem.

To design the optimal filter, we obtain the variation in P_e at $h(t)$ in the direction $\delta h(t)$,

$$\delta P_e(h, \delta h) = \frac{E_{s_0}[e^{-h*s_0}(h * s_0)^F(\delta h * s_0)]}{F!} - \frac{E_{s_1}[e^{-h*s_1}(h * s_1)^F(\delta h * s_1)]}{F!}, \tag{6}$$

where $F = \lfloor \gamma \rfloor$ and $*$ signifies continuous time convolution. One method of obtaining an optimal filter is to equate (6) to zero and solve for $h(t)$ which satisfies this for all $\delta h(t)$. Trivially this equation is satisfied by $h(t) = 0\ \forall t$, yielding a maximum probability of error. Solving $\delta P_e(h, \delta h) = 0\ \forall \delta h(t)$ is equivalent to solving

$$E_{s_0}[s_0(t)e^{-h*s_0}(h * s_0)^F] - E_{s_1}[s_1(t)e^{-h*s_1}(h * s_1)^F] = 0 \quad \forall t, \tag{7}$$

which in most cases has no solution. In any case the filter $h(t)$ which satisfies this equation may not satisfy the inequality constraint (5).

Since the minimizing filter is not evident, let us examine the expression for P_e further. The second variation of P_e at $h(t)$ in the direction $\delta h(t)$ is

$$\delta^2 P_e(h, \delta h) = \frac{E_{s_0}[(\delta h * s_0)^2 e^{-h*s_0}(h * s_0)^{F-1}(h * s_0 - F)]}{F!} - \frac{E_{s_1}[(\delta h * s_1)^2 e^{-h*s_1}(h * s_1)^{F-1}(h * s_1 - F)]}{F!}. \tag{8}$$

It is clear from this last expression that if the threshold F is odd[1] and such that

$$h * s_0 \leq F \leq h * s_1 \quad \text{almost surely}, \tag{9}$$

then the probability of error is concave everywhere and in every direction $\delta h(t)$, and the optimal filter must achieve some inequality constraint at every t with equality. This shows that under this condition on the threshold, the minimum probability of error optical filter has impulse response $h(t) \in \{0, 1\}\ \forall t$ according to constraint (5). The exact form of the filter then allows certain time periods to pass unaltered and removes others altogether, which is easily implemented using a tapped delay line.

The condition on the threshold F given by (9) is not always satisfied unless decision feedback is used to regulate the threshold. If the uncertainty is removed from the received intensities $s_0(t)$ and $s_1(t)$ by providing for instance estimates of subsequent decisions in an ISI environment, then the threshold can be set to satisfy the condition. This condition is assumed for the remainder of the paper.

It is still unclear how to determine what the optimal filter function should be. The difficulty is the constraint, which must be finitely approximated. To this end, let us break up the received signal into L equal length intervals, and let the signals and filter be represented by L-vectors $\underline{s}_i$ and $\underline{h}$ respectively. As L increases, the piecewise constant filter approximates the desired continuous time filter $h(t)$. We can write the cost function to minimize as

$$J(\underline{h}) = E\left[\sum_{n=0}^{\lfloor \gamma \rfloor} \frac{1}{n!} e^{-\underline{h}^T \underline{s}_0}(\underline{h}^T \underline{s}_0)^n + \sum_{n=\lceil \gamma \rceil}^{\infty} \frac{1}{n!} e^{-\underline{h}^T \underline{s}_1}(\underline{h}^T \underline{s}_1)^n\right] + \underline{m}^T \underline{h}, \tag{10}$$

[1]Even if the threshold F is not odd, the filter response and the received signal must both be nonnegative, so the argument is still valid.

where $\underline{m}$ is the vector of Lagrange multipliers and the superscript T indicates transposition. The vector $\underline{\tilde{h}}$ has component j equal to $(h_j - 1/2)^2$, which allows for a differentiable constraint. Differentiating and equating to zero yields

$$E_{\underline{s}_0}[s_{j0}e^{-\underline{h}^T\underline{s}_0}(\underline{h}^T\underline{s}_0)^F] - E_{\underline{s}_1}[s_{j1}e^{-\underline{h}^T\underline{s}_1}(\underline{h}^T\underline{s}_1)^F] + 2F!m_j(h_j - \frac{1}{2}) = 0; \quad j = 1, 2, \ldots, L, \quad (11)$$

where s_{ji} and m_j are the jth element of vectors $\underline{s}_i$ and $\underline{m}$ respectively.

This equation is not quickly solved, even numerically. The solution relies on an L dimensional optimization. Fortunately, in application some insight into the problem greatly simplifies the problem. It is evident that the filter will pass certain sections of the received bit that have low noise or low distortion, and perhaps block others that have too high a distortion for the amount of information provided. Using this type of argument, the optimization problem results in simply finding the segments of the received bit the filter will pass unchanged and the segments the filter will block. This method is illustrated in Section 4.

3.2 Nonideal Photodetection

If a bandlimited photodetector is used, there is no simple closed-form expression for the probability of error, since the observation X is now an integrated filtered Poisson process. A good upper bound to the error probability is the Chernoff bound, which does have a closed-form expression in this case. The Chernoff bound can be written as

$$P_e \leq \text{CB} = E\left[\frac{1}{2}\exp\{\mu_0(x) - xF\} + \frac{1}{2}\exp\{\mu_1(-x) + xF\}\right] \quad \forall\, 0 \leq x \leq 1, \quad (12)$$

where $\mu_i(x)$ is the logarithm of the moment generating function under hypothesis i, and the expectation is again taken over the random arrival rates λ_0 and λ_1. For an integrated filtered Poisson process, the moment generating function yields

$$\mu_i(x) = \int_0^T \int_{-\infty}^t h(t-\tau)s_i(\tau)\, d\tau (e^{xg(T-t)} - 1)\, dt, \quad (13)$$

where $g(t)$ is the impulse response of the photodetector and $h(t)$ is the impulse response of the fiber-optic filter as before.

In this case, we can again take the second variation of the cost function at $h(t)$ in every direction $\delta h(t)$, yielding

$$\delta^2\text{CB}(h, \delta h) \propto E\left[e^{\mu_1(-x)+Fx}\left\{\int_0^T \delta h * s_1(e^{-xg(T-t)} - 1)\, dt\right\}^2 + e^{\mu_0(x)-Fx}\left\{\int_0^T \delta h * s_0(e^{xg(T-t)} - 1)\, dt\right\}^2\right]. \quad (14)$$

This time we can see that the second variation in everywhere *positive*, so we can make no assumptions about the existence of a minimum inside the inequality constraints. The convexity is a direct consequence of the exponential form of this bound, and does not imply that the actual probability of error is convex.

As before, we need to make a vector approximation to solve the problem with the pointwise inequality constraint. This approximation now yields an $L+1$ dimensional optimization problem, since we must optimize over $x \in [0,1]$ also. We can no longer use our insight into a specific problem to simplify the calculation, since we are not simply identifying regions of integration, but have to find the real values of the impulse response at every t. The solution is therefore very computationally intensive to derive.

4 Applications to a Dispersive Channel

The filter design methods discussed above are applied to a dispersive channel suffering from intersymbol interference (ISI). Both the ideal photodetection and nonideal photodetection cases are examined. The received signal for a general ISI channel can be modeled as

$$s(t) = \sum_{k=-\infty}^{\infty} b_k \tilde{s}(t - kT), \tag{15}$$

where $\tilde{s}(t)$ is a deterministic and known signal of duration greater than T, and b_k is the bit transmitted during the time interval $(kT, (k+1)T]$. Assuming no more than one bit spread, we can simplify (15) and write the rate function after filtering for $t \in (0, T]$ as

$$\lambda_{b_0}(t) = b_0 \int_{-\infty}^{t} h(t-\tau)\tilde{s}(\tau)\, d\tau + b_{-1} \int_{-\infty}^{t} h(t-\tau)\tilde{s}(\tau+T)\, d\tau + \lambda_d. \tag{16}$$

For example, let us consider a received signal which is constant over $(0, T]$ and decreases exponentially for $t > T$,

$$\tilde{s}(t) = \begin{cases} \lambda & 0 < t \leq T, \\ \lambda e^{-\alpha(t-T)} & T < t < 2T, \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

The exponential drop-off region will contaminate the following bit. The severity of the ISI is determined by the parameter α. This is the model used for filter design in the following sections.

4.1 ISI Channel with Ideal Photodetection

In the case of ideal photodetection, equation (11) can be used to design the optimal filter. As was alluded to before, it is often easier to consider the problem at hand and obtain a simplified model instead. The simplified parameter of this Poisson random variable under hypothesis $b_0 = i$ is given by

$$\Lambda_i = \int_0^T \lambda(ih(T-t) + b_{-1}h(T-t)e^{-\alpha t})\, dt + T\lambda_d. \tag{18}$$

Since the filter $h(t)$ has value either 1 or 0 at each time t, we can simply consider the integration interval for the accumulation of Λ_i. It is easy to imagine that the filter would exclude the very beginning of the bit period when ISI is strongest, if at all. Therefore, the filter has the form

$$h(t) = \begin{cases} 0 & t > T^* \\ 1 & 0 \leq t \leq T^*, \end{cases} \tag{19}$$

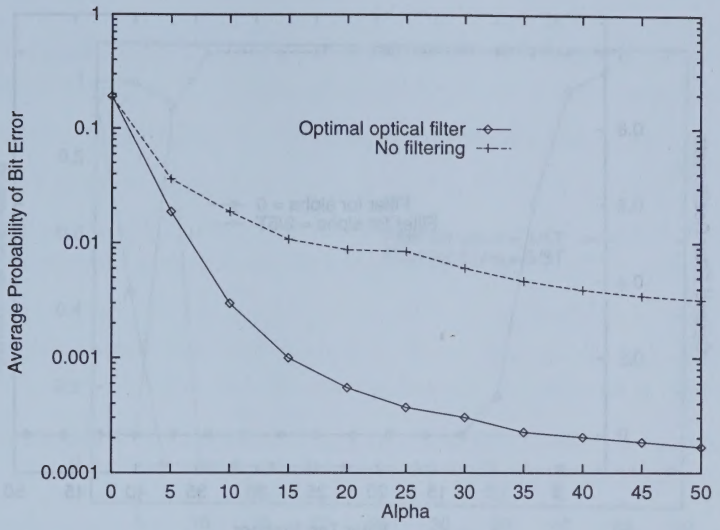


Figure 2: Probability of error as a function of the parameter α representing the amount of dispersion in the binary communication channel. In figure $\alpha = 1$ corresponds to an actual $\alpha = 1/200T$ in equation (18).

for some time $T^* \in (0, T]$. For small α, some decision feedback is assumed in the electronic domain so that the threshold γ is optimized and condition (9) holds.

We can minimize the probability of error over T^* directly, yielding a solution much faster than solving (11) directly, since the optimization is only over one variable. Taking a derivative of P_e with respect to T^* and equating to zero in this case results in

$$(T - T^*)^F e^{-\lambda(T-T^*)} + \left(T - T^* + \frac{1}{\alpha}(e^{-\alpha T^*} - e^{-\alpha T})\right)^F \exp\left\{-\lambda(T - T^* + \frac{\lambda}{\alpha}(e^{-\alpha T^*} - e^{-\alpha T})\right\} e^{-\alpha T^*} - \left(\frac{1}{\alpha}(e^{-\alpha T^*} - e^{-\alpha T})\right)^F \exp\left\{-\frac{\lambda}{\alpha}(e^{-\alpha T^*} - e^{-\alpha T})\right\} e^{-\alpha T^*} = 0, \quad (20)$$

which can be solved iteratively for T^*.

The performance of the filters can be calculated directly using (4) for the chosen filter. For the dispersive channel described above, the error probability as a function of α, the decay rate of the dispersed signal, is plotted in Figure 2. The probability of error without the filter is also plotted. It is easily seen that the filtering improves the error probability considerably in this simple case.

4.2 ISI Channel with Nonideal Photodetection

Consider a dispersive optical channel with received waveforms (17). We will optimize the filter impulse response my minimizing (12) for two different photodetector models.

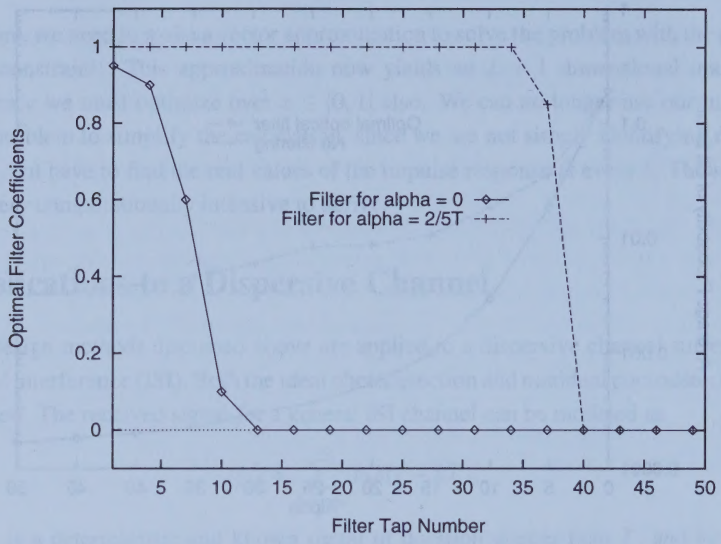


Figure 3: Optimal filter for several α values for an exponentially decaying photodetector impulse response with parameter $\beta = 1/T$.

The first model is a photodetector with an exponential decay for an impulse response,

$$g(t) = \begin{cases} e^{-\beta t} & t \geq 0 \\ 0 & \text{otherwise,} \end{cases} \tag{21}$$

where β determines the bandwidth of the photodetector. Through a 16 dimensional optimization we find the resulting filter, which is plotted in Figure 3 for $\alpha = 0$ and $\alpha = 2/5T$. The optimal all-positive filter response in this case is not strictly 0-1, as in the ideal case. The form is very similar, except that there is a slower transition from the signal pass $h(t) = 1$ values to the signal block $h(t) = 0$ values.

The second photodetector impulse response tested is the typical raised cosine function

$$g(t) = \begin{cases} \frac{1}{2} - \frac{1}{2}\cos(2\pi t\beta) & 0 \leq t \leq \frac{1}{\beta} \\ 0 & \text{otherwise,} \end{cases} \tag{22}$$

where β again determines the bandwidth of the photodetector. The resulting optimal filter is shown in Figure 4. In this case also the optimal filter is nearly a 0-1 filter as in the ideal photodetector case. The same slopped transitions are seen as in the negative exponential case discussed above.

5 Conclusions

A method of designing optical filters for fiber-optic communications is proposed. These filters are designed to minimize the probability of error under the constraint that the filter response is nonnegative and less than unity for all time. They can be implemented optically using weighted

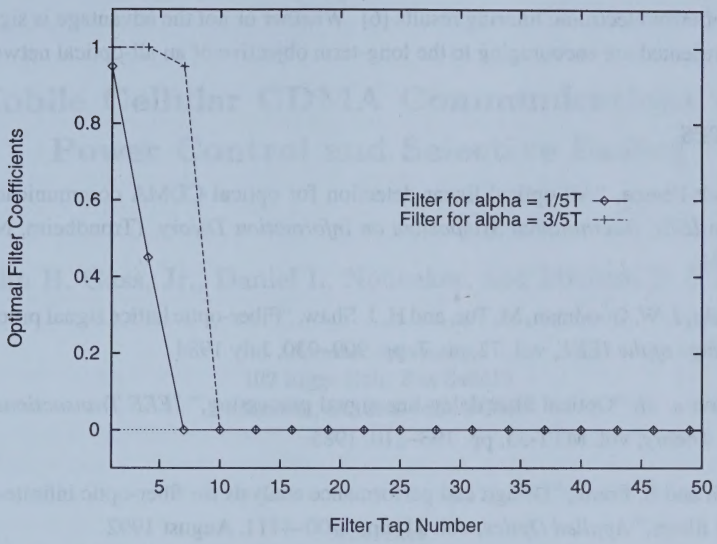


Figure 4: Optimal filter for several α values for an raised cosine photodetector impulse response with parameter $\beta = 1/2T$.

fiber-optic tapped-delay lines. Two cases are considered: an ideal infinite bandwidth impulse response photodetector versus a bandlimited photodetector. The filter designs are applied to a dispersive ISI channel.

For the case of an ideal photodetector, there exists a simple closed-form expression for the error probability which can be minimized directly. Applying calculus of variations to this optimization problem indicates that, under some conditions on the threshold, the error probability is concave everywhere and in every direction. This proves that the optimal filter will achieve some inequality constraint with equality. The filters considered are are filters with impulse response between zero and one for all time, and thus under this constraint, the optimal filters have impulse response exactly zero or one at each time t. This allows an implementation using unweighted tapped delay lines, or simple control on the photon accumulation interval. A communication through a dispersive ISI channel using the optimal filter is found to perform considerably better than without the filter.

For the case of a bandlimited photodetector response, there is no simple closed-form expression for the error probability, but there is a simple upper bound, namely the Chernoff bound, which can be optimized instead. The second variation of the Chernoff bound with respect to the filter is found to be nonnegative everywhere, and thus a minimum can exist inside the constraint and the filter must not necessarily be 0-1. A vector form of the filter must be optimized to include the pointwise inequality constraint. When this method is applied to a dispersive channel, the resulting filter is approximately 0-1 with a slow transition.

This work has shown that it is advantageous to filter an information signal in the optical domain. Further work is needed to show whether any real advantage exists in filtering optically instead of electronically. This can be done by comparing the results presented here to minimum

probability of error electronic filtering results [6]. Whether or not the advantage is significant, the results presented are encouraging to the long-term objective of an all-optical network.

References

[1] M. Brandt-Pearce, "All-optical linear detection for optical CDMA communication systems," in *IEEE International Symposium on Information Theory*, (Trondheim, Norway), June 1994.

[2] B. Moslehi, J. W. Goodman, M. Tur, and H. J. Shaw, "Fiber-optic lattice signal processing," *Proceedings of the IEEE*, vol. 72, no. 7, pp. 909–930, July 1984.

[3] K. Jackson *et al*, "Optical fiber delay-line signal processing," *IEEE Transactions on Microwave Theory*, vol. MTT-33, pp. 193–210, 1985.

[4] A. Ghosh and S. Frank, "Design and performance analysis for fiber-optic infinite-impulse response filters," *Applied Optics*, vol. 31, pp. 4700–4711, August 1992.

[5] P. Paparao, A. Ghosh, and S. D. Allen, "Design and performance optimization of fiber optic adaptive filters," *Applied Optics*, vol. 30, pp. 1826–1838, May 1991.

[6] J. R. F. Da Rocha and J. J. O'Reilly, "Linear direct-detection fiber-optic receiver optimization in the presence of intersymbol interference," *IEEE Trans. on Comm.*, vol. 34, no. 4, pp. 365–374, April 1986.

Mobile Cellular CDMA Communications with Power Control and Selective Fading *

John H. Gass, Jr., Daniel L. Noneaker, and Michael B. Pursley

Clemson University
102 Riggs Hall, Box 340915
Clemson, SC 29634-0915 USA

Abstract

The bit error probability is determined for a direct-sequence code-division multiple-access (CDMA) communications system which employs closed-loop power control of the transmitted signal in the presence of doubly selective fading. Several selective fading channels are employed as examples to illustrate the dependence of system performance on the characteristics of the channel, the number of taps in the rake receiver, and the chip rate of the direct-sequence waveform. The influence of the channel delay spectrum and Doppler spread on the effectiveness of the power-control method is examined, and the sensitivity of the performance of the power control to the chip rate and the number of taps is discussed.

I. Introduction

Direct-sequence (DS) spread spectrum has been proposed for use in mobile cellular code-division multiple-access (CDMA) systems [1, 2] and personal communication systems and networks [3, 4]. Important characteristics of a CDMA system include the chip rate [2, 3], the receiver structure and complexity, and the power-control technique. The purpose of this paper is to focus on the power control of a CDMA system and examine its effect on the performance of systems with different chip rates and different receiver structures.

A multiple-access system that employs DS spread-spectrum modulation is susceptible to *near-far interference* [5]. Near-far interference occurs when the receiver input includes

*This research was supported in part by the Holcombe Endowment at Clemson University and in part by the Army research Office under grants DAAH04-94-G-0154 and DAAH04-93-G-0253. J. H. Gass is the recipient of a National Science Foundation Graduate Research Fellowship.

one or more other CDMA signals that are stronger than the desired signal. The near-far effect can be reduced by adapting the power of each transmitter to the channel response or the interference environment. This technique is referred to as *power control* [6]. The reverse (mobile to base station) link of a cellular CDMA system is particularly susceptible to near-far interference, though inter-cell interference can cause near-far interference on the forward link as well. In this paper, we consider power control that is used to reduce intra-cell near-far interference on the reverse link. Power control does not completely eliminate the near-far effect, because the control is based on imperfect estimates of the strength of the received signals and there is some delay before the transmitter power can be adjusted. The amount of residual variation in the strength of each received signal depends on the technique that is used for power control, the rate of variation of the channel response, the chip rate of the system, and the type of receiver that is employed.

For a multipath fading channel, DS spread spectrum provides a signal waveform that permits an appropriately designed receiver to resolve some or all of the multipath components of the received signal. DS spread spectrum with a sufficiently high chip rate permits discrimination against unwanted multipath components in a standard correlation receiver or recombination of these components in a rake receiver [7]. The probability of error is determined in [8] for the demodulator output in a CDMA system that employs binary differential phase-shift keying (DPSK), noncoherent demodulation, and a rake receiver with equal-gain square-law combining. The performance of this system is shown to depend on both the characteristics of the channel and the chip rate of the system. The system analyzed in [8] uses an average power control that compensates only for long-term variations in the net path loss.

In this paper, we extend the investigation in [8] to consider a system that uses rapid power control in an attempt to compensate for short-term variations in the strength of the received signal. We examine the effectiveness of closed-loop power control for systems of different chip rates and with different numbers of taps in the rake receiver. As in [8], a CDMA system with a normalized chip rate of 50 chips per symbol is taken as a representative low-chip-rate system, and a chip rate of 400 is used as an example of a high-chip-rate system.

II. Doubly Selective Fading Channels

The mobile communications channel often consists of a few strong paths (perhaps only one) and a number of weaker paths [3]. For a DS system with a low chip rate, the delays for multiple multipath components fall within one chip interval. The composite signal formed by such components may be modeled as a single diffuse fading signal that has a Rician fading amplitude with a strong Rayleigh component and a relatively weak specular component. However, if the DS signal has a chip interval that is much smaller than the smallest difference among the delays for the various propagation paths, a correlation receiver can lock onto a single multipath component. Thus, the channel may appear to the correlation receiver to produce a single component that exhibits Rician fading with a strong specular component and a relatively weak Rayleigh component.

We model the channel in a manner that reflects the phenomenon described above and allows for tractable analysis of the receiver. The channel model consists of several

clusters of paths, with each cluster having a single specular path plus a continuum of Rayleigh-fading paths with path delays that are centered at the delay of the specular path.

This channel is a special case of the Gaussian wide-sense-stationary uncorrelated-scattering (WSSUS) channel [9]. It is represented as a complex Gaussian random process $h(t, \xi)$, which is characterized by its specular part

$$E[h(t,\xi)] = \sum_{i=1}^{M} \rho_i \delta(\xi - \xi_i)$$

and the autocovariance function of its diffuse part

$$Cov[h(t,\xi), h(x,\alpha)] = \sum_{i=1}^{M} 2\sigma_i^2 d(t-x) g_i(\xi) \delta(\alpha - \xi),$$

where t and x are time variables and ξ and α are path delay variables. The number of path delay clusters in the channel is M. The parameter ρ_i is the amplitude of the ith specular component, d is the *time-correlation function* of the channel, and g_i is the *delay spectrum* of the ith cluster of paths.

For convenience we assume that ρ_i is real. In our examples we consider rectangular delay spectra, so that the delay spectra are defined by

$$g_i(\xi) = \begin{cases} 1/(\mu_i T), & |\, \xi - \xi_i \,| < \mu_i T/2, \\ 0, & otherwise, \end{cases}$$

where T is the data pulse interval and μ_i is the normalized *total delay spread* of the ith cluster. The total delay spread of the ith cluster is the support of the function g_i. It is assumed that the support of g_i and the support of g_j are disjoint for distinct i and j. The time-correlation function d is characterized by the half-power bandwidth B_d of the Fourier transform of d, and the normalized half-power bandwidth $D_T = B_d T$ is referred to as the *Doppler spread* of the channel [10]. In our examples we consider exponential time-correlation functions. Another parameter of interest is the *energy ratio*, defined by $\zeta_i = \rho_i^2/(2\sigma_i^2)$, which is the ratio of the specular energy to the average diffuse energy in the components of the received signal associated with the ith cluster of paths.

III. Channel Estimation and Power Control

In closed-loop power control, the base station obtains an estimate of some characteristic of the signal that is received from the mobile unit. Possible estimated quantities include the signal energy and the signal-to-interference ratio at the output of a correlator or rake receiver [11]. The estimate is compared with a target value, the receiver determines the change in transmitter power that is needed to meet the target, and a command to increase or decrease the transmitter power by the appropriate amount is sent to the mobile unit via a control channel. Thus, estimates in closed-loop power control are delayed before they are used, and so they are susceptible to inaccuracies caused by any rapid variations in the channel. In addition, the estimates are subject to inaccuracy due to multiple-access interference and thermal noise at the receiver. If the control commands

are transmitted to the mobile unit without sufficient error protection, they are subject to channel errors. In this investigation, we consider an estimator that produces a perfect estimate of the signal energy at the output of a rake receiver. The corresponding power-control command is received at the mobile unit with some fixed delay, so that it is subject to inaccuracy due to channel variations. We assume that any channel errors that occur when the base station transmits the command to the mobile unit are corrected by forward error-correction coding.

The system considered in this paper uses estimates of the signal component at the output of each tap of the rake receiver. Let $a_{i,k}$ denote the gain of the channel at the ith tap output during the kth data pulse interval. If the baseband data pulse in the kth data pulse interval has polarity b_k, the amplitude of the baseband signal component at the output of the ith tap during that interval is given by $b_k\, a_{i,k}\, \mathcal{E}_k^{1/2}$, where $\mathcal{E}_k$ is the transmitted energy during the kth data pulse interval. Without loss of generality we consider an estimate of the signal amplitude during the zeroth data pulse interval. An estimate of $a_{i,0}\, \mathcal{E}_0^{1/2}$, which is denoted $\hat{a}_i \mathcal{E}_0^{1/2}$, is formed at the receiver. For convenience we refer to $\hat{a}_i$ as the estimate of the channel gain at the output of the ith tap, though only $\hat{a}_i \mathcal{E}_0^{1/2}$ is explicitly estimated. From the estimates of the signal components the receiver forms an estimate of the total signal energy at the output of the L taps and compares the energy estimate with the target value $\mathcal{E}_{target}$. A command is sent to the mobile unit to change its level of transmitted power by a multiplicative factor Δ that is given by

$$\Delta = \mathcal{E}_{target} / (\|\hat{a}\|^2 \mathcal{E}_0),$$

where

$$\hat{a} = (\hat{a}_1, \cdots, \hat{a}_L)$$

and

$$\|\hat{a}\|^2 = \sum_{i=1}^{L} |\, \hat{a}_i \,|^2 \,.$$

The estimator that we consider in this paper is perfect, so that $\hat{a}_i$ is equal to $a_{i,0}$, and the distribution function of the random vector $\hat{\mathbf{a}}$ is determined by the parameters of the channel, the chip rate, and the number of taps in the rake receiver.

Both $\mathcal{E}_{target}$ and $\mathcal{E}_0$ are known at the mobile unit, so that it can determine $\|\hat{a}\|^2$ from Δ. The value $\|\hat{a}\|^2$ is used by the power-control algorithm in the mobile unit to determine the transmitted energy during the mth and $(m+1)$th data pulse intervals, where m is the delay in data pulse intervals in the use of the estimate. If $\mathcal{E}_{max}$ is the maximum energy per data pulse that the transmitter can produce and

$$\mathcal{E}_{target} = C\, \mathcal{E}_{max},$$

the transmitted energy determined by the power-control algorithm is given by

$$\mathcal{E}(\hat{a}) = (\|\hat{a}\|^2 + C)^{-1}\, \mathcal{E}_{target}.$$

IV. Error Probabilities for the Rake Receiver

The receiver that we consider is the post-detection equal-gain square-law diversity combining rake receiver. The ith tap of the rake receiver is synchronized with the received

signal component arriving with path delay ξ_i. The receiver employs differentially coherent demodulation, and it makes one bit decision based on the signal received over two consecutive data pulse intervals. Suppose that $(U_{i,m}, U_{i,(m+1)})$ are the sampled matched-filter outputs at the ith receiver tap during the mth and $(m+1)$th pulse intervals, respectively. The receiver makes a binary decision based on the sign of the real-valued decision statistic

$$Z_l = \sum_{i=1}^{L} (U_{i,(m+1)} U_{i,m}^* + U_{i,(m+1)}^* U_{i,m}),$$

where L is the number of receiver taps.

The analysis can be simplified by modeling the multipath interference as Gaussian noise and by assuming that the aperiodic autocorrelation function of the spreading sequence is ideal. For good spreading sequences and a small number of taps, this assumption of an ideal autocorrelation function has minimal effect on the determination of the bit error probability [12]. A rectangular chip waveform is considered for the DS signal, and the chip duration is denoted by T_c. The composite interference, which includes the multiple-access interference, multipath interference, and thermal noise, is modeled as additive white Gaussian noise with single-sided power spectral density N_0. The average signal-to-interference ratio *for the specular component* of the received signal *due to the first cluster* is defined as

$$SIR = E[\rho_1^2 \mathcal{E}(\hat{\mathbf{a}})/N_0], \tag{1}$$

where the expectation is with respect to the random estimate vector $\hat{\mathbf{a}}$. Without loss of generality, we assume that $N_0 = 1$. Under some circumstances, the value of SIR that is required to achieve a specified probability of error is inversely proportional to the capacity of the system per unit bandwidth. One such circumstance occurs if the mobile transmitters are subject to independent, identically-distributed channel processes, each mobile unit employs the same power-control algorithm, and the performance of the system is limited by multiple-access interference.

The L estimates are independent random variables and $\hat{\mathbf{a}}$, $\{\mathbf{a}_{i,m}\}$, and $\{\mathbf{a}_{i,m+1}\}$ are jointly Gaussian random variables. The variance of $\hat{\mathbf{a}}_i$ is denoted by $\hat{\sigma}_i^2$, and the covariance of $\hat{\mathbf{a}}_i$ and $\mathbf{a}_{i,k}$ is denoted by $\hat{c}_{i,k}^2$ for $k = m$ and $k = (m+1)$. As discussed in [12], an accurate approximation of the effect of time selectivity on the error probability can be obtained by ignoring its effect within a pulse interval. Thus if

$$\alpha_i = \sqrt{\min\{\mu_i - (T\mu_i^2/2T_c) + (T^2\mu_i^3/12T_c^2), 2T_c/3T\}/\mu_i},$$

where T_c is the chip interval, the variance of $\hat{\mathbf{a}}_i$ is given by

$$\hat{\sigma}_i^2 \approx 2\alpha_i^2 \sigma_i^2,$$

and the covariance of $\hat{\mathbf{a}}_i$ and $\mathbf{a}_{i,k}$ is given by

$$\hat{c}_{i,k}^2 \approx 2\alpha_i^2 \sigma_i^2 \; d(mT)$$

for $k = m$ and $k = (m+1)$.

Conditioned on $\hat{\mathbf{a}}$ and on b_m and $b_{(m+1)}$, the polarity of the mth and $(m+1)$th data pulses, the sampled output of the ith tap in the kth pulse interval is normally distributed with a mean value of

$$b_k \overline{\rho}_i = b\mathcal{E}^{1/2}(\hat{a}) \left[\rho_i + \hat{c}_{i,k}^2 (\hat{a}_i - \rho_i)/\hat{\sigma}_i^2\right]$$

and a variance of

$$2\overline{\sigma}_i^2 + 1 = \mathcal{E}(\hat{a}) \left[2\alpha_i^2 \sigma_i^2 - \hat{c}_{i,k}^4 / \hat{\sigma}_i^2 \right] + 1,$$

for $k = m$ and $k = (m+1)$, and the covariance of the sampled outputs of the ith tap in the mth and $(m+1)$th pulse intervals is given by

$$Cov(U_i, V_i | \hat{\mathbf{a}}) \approx 2 b_{-1} b_0 d(T) \overline{\sigma}_i^2.$$

The average probability of error is given by

$$\overline{P}_b = E[P_b(\hat{\mathbf{a}})], \tag{2}$$

where $P_b(\hat{\mathbf{a}})$ is the probability of error conditioned on $\hat{\mathbf{a}}$. If the channel is doubly selective and the $\overline{\sigma}_i$ are distinct for $1 \leq i \leq L$, the average probability of error conditioned on $\hat{\mathbf{a}}$ is given by

$$P_b(\hat{\mathbf{a}}) = \sum_{j=L+1}^{2L} A_j \prod_{i=1}^{L} \frac{\tilde{\sigma}_j^2}{\tilde{\sigma}_j^2 + \tilde{\sigma}_i^2} \exp[-2\overline{\rho}_i^2 / (\tilde{\sigma}_j^2 + \tilde{\sigma}_i^2)], \tag{3}$$

where

$$A_j = \prod_{i \geq L+1, i \neq j} \frac{\tilde{\sigma}_j^2}{\tilde{\sigma}_j^2 - \tilde{\sigma}_i^2}$$

and

$$\tilde{\sigma}_i^2 = \begin{cases} 2(1 + d(T))\overline{\sigma}_i^2 + 1 & 1 \leq i \leq L \\ 2(1 - d(T))\overline{\sigma}_{i-L}^2 + 1 & L < i \leq 2L \end{cases}.$$

If the $\overline{\sigma}_i$ are not all distinct, the probability of error is given by an expression that is obtained in the same manner as (3).

V. Results and Conclusions

In the numerical results that are presented, we consider parameter values that are appropriate for two CDMA systems of practical interest. The first is a digital cellular system operating at 900 MHz with a channel bit rate of 25 kbps. The second is a PCS system operating at 2 GHz, and it also has a bit rate of 25 kbps. The Doppler spreads that are considered in the examples are shown in Table 1 along with the corresponding speed of the mobile unit for either system. The feedback delays that are considered for the power-control commands are given in Table 2. Two channels are considered in the examples. Channel A consists of three identically distributed clusters, and each cluster has a specular component of amplitude ρ and an energy ratio of ζ. Neither the second nor the third cluster of channel B has a specular component ($\rho_2 = \rho_3 = 0$), and the second and third clusters each have a diffuse component with an average power that is 6 dB less than that of the diffuse component of the first cluster ($\sigma_2 = \sigma_3 = \sigma/2$). The delay profiles of both channels are shown in Figure 1.

The normalized chip rate of the CDMA system is denoted R, and it is given by

$$R = T/T_c.$$

Chip rates of 50 and 400 are chosen as representative low and high chip rates, respectively. For a system with a bit rate of 25 kbps, a normalized chip rate of 50 corresponds to 1.25

900 MHz		2 GHz	
D_T	Speed (kph)	D_T	Speed (kph)
0.00025	7.5	0.00018	2.5
0.0005	15	0.00037	5
0.002	60	0.0011	15
0.004	120	0.0044	60

Table 1: Values of D_T and the corresponding speeds for a 25 kbps system.

m	Delay (ms)
31.25	1.25
62.5	2.5
125	5.0
250	10.0

Table 2: Values of m and the corresponding delays for a 25kbps system.

Mchips/sec. and a normalized chip rate of 400 corresponds to 10 Mchips/sec. Figures 2–4 show the SIR required to achieve a bit error probability of 10^{-2} as a function of the chip rate for different Doppler spreads. For the low-chip-rate system, there is a significant degradation in performance as the Doppler spread increases. The sensitivity to Doppler spread with a low chip rate is very pronounced with a correlation receiver, and it is shown in Figure 2. The sensitivity is reduced if a three-tap rake receiver is employed, as seen by comparing Figure 2 with Figures 3 and 4. In addition, the weak signal components captured by the second and third taps for channel B result in a greater degradation in the performance of the rake receiver with large Doppler spreads for channel B than for channel A. Figure 5 shows that the performance of the low-chip-rate system and a correlation receiver is also very sensitive to the feedback delay in the power-control command. The use of rake reception reduces the sensitivity but a large delay significantly degrades performance, even with a three-tap rake receiver. These results demonstrate that the effectiveness of rapid power control for a low-chip-rate system in the presence of fading is highly dependent on the timeliness of the estimate.

In contrast, Figures 2–4 illustrate that the performance of the high-chip-rate system shows little sensitivity to the Doppler spread with either a correlation receiver or a rake receiver for either channel A or channel B. Similarly, Figures 5–7 illustrate that the high-chip-rate system is not very sensitive to the feedback delay in the power-control command. Furthermore, the rake receiver provides better performance than the correlation receiver for the high-chip-rate system and channel A, but it yields poorer performance than the correlation receiver for channel B. Thus the high-chip-rate system benefits from rake reception only if each tap resolves a strong specular component.

Because of the ability of the high-chip-rate system to resolve a stable specular component with a correlation receiver, the high chip rate is superior to the low chip rate for correlation reception and a wide range of channel parameters and feedback delays. This is illustrated in Figures 2 and 5. In contrast, the low chip rate provides better performance with a rake receiver than the high chip rate if the Doppler spread and feedback delay are sufficiently small for the power control of the low-chip-rate system to adapt to changes in the energy of the received signal. This is seen in Figures 3,4,6, and 7. However if the channel changes more quickly than the ability of the low-chip-rate system to respond

to the changes, the high-chip-rate system with a correlation receiver provides superior performance to the low-chip-rate system with a rake receiver.

References

[1] K. S. Gilhousen, I. M. Jacobs, R. Padovani, A. J. Viterbi, L. A. Weaver, Jr., and C. E. Wheatley III, "On the capacity of a cellular CDMA system," *IEEE Trans. Veh. Technol.*, vol. VT-40, pp. 303–312, May 1991.

[2] IS-95, "Mobile station – base station compatability standard for dual-mode wideband spread spectrum cellular system," *Telecommunications Industry Association/ Electronics Industry Assocation Interim Standard*, July 1993.

[3] D. L. Schilling and L. B. Milstein, "Broadband CDMA for indoor and outdoor personal communication," *Proc. of the Fourth Internat. Symp. on Personal, Indoor, and Mobile Commun.*, Sept. 1993.

[4] R. L. Pickholtz, L. B. Milstein, and D. L. Schilling, "Spread spectrum for mobile communications," *IEEE Trans. Veh. Technol.*, vol. VT-40, pp. 313–322, May 1991.

[5] M. B. Pursley, "The role of spread spectrum in packet radio networks," *Proc. IEEE*, vol. 75, pp. 116–134, Jan. 1987.

[6] A. J. Viterbi, A. M. Viterbi, and E. Zehavi, "Performance of power-controlled wideband terrestrial digital communication," *IEEE Trans. Commun.*, vol. COM-41, pp. 559–569, Apr. 1993.

[7] R. Price and P. E. Green, "A communication technique for multipath channels," *Proc. IRE*, vol. 46, pp. 555–570, March 1958.

[8] D. L. Noneaker and M. B. Pursley, "On the chip rate of CDMA systems with doubly selective fading and rake reception," *IEEE Journal on Selected Areas in Communications*, vol. 12, pp. 853–861, June 1994.

[9] P. A. Bello, "Characterization of randomly time variant linear channels," *IEEE Trans. Commun. Syst.*, vol. COM-11, pp. 360–393, Dec. 1963.

[10] P. A. Bello and B. D. Nelin, "The influence of fading spectrum on the binary error probabilities of incoherent and differentially coherent matched filter receivers," *IEEE Trans. Commun. Syst.*, vol. CS-10, pp. 160–168, June 1962.

[11] S. Ariyavistakul and L. F. Chang, "Signal and interference statistics of a CDMA system with feedback power control," *IEEE Trans. Commun.*, vol. COM-41, pp. 1626–1634, Nov. 1993.

[12] D. L. Noneaker and M. B. Pursley, "The effects of spreading sequence selection on DS spread-spectrum with selective fading and two forms of rake reception," *Proc. 1992 Global Telecommunications Conference*, vol. 4, pp. 66–70, Dec. 1992.

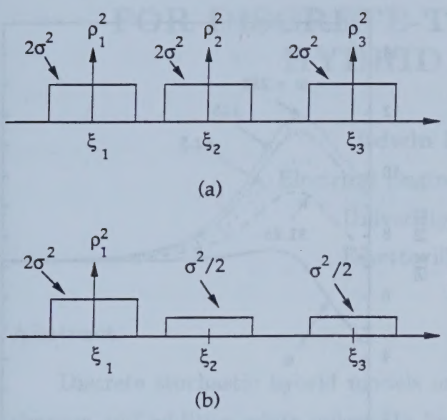


Figure 1. Delay spectra of channels A and B ($\mu_1 = \mu_2 = \mu_3 = 0.05$, $\zeta_1 = -3$ dB). (a) Channel A. (b) Channel B.

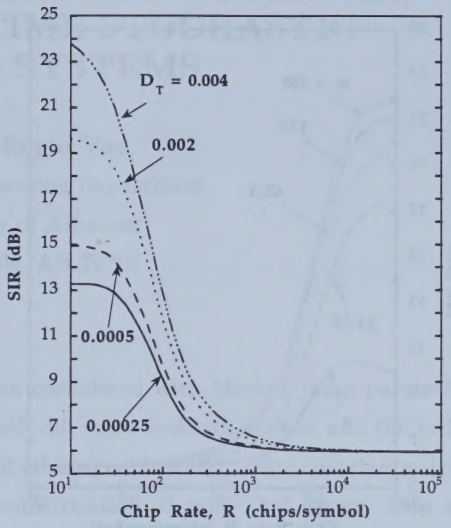


Figure 2. Required SIR for $\overline{P}_b=10^{-2}$ for Channel B, estimate delay of 62.5, and L = 1.

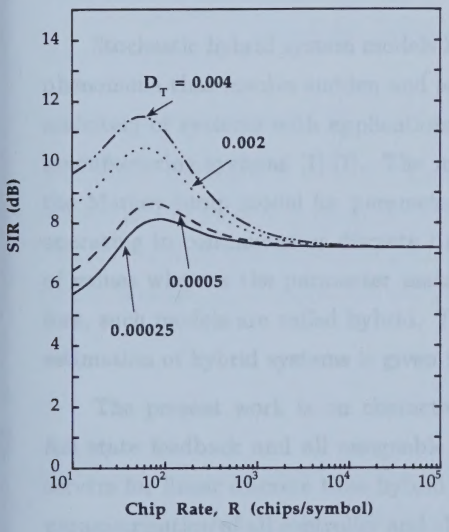


Figure 3. Required SIR for $\overline{P}_b=10^{-2}$ for Channel B, estimate delay of 62.5, and L = 3.

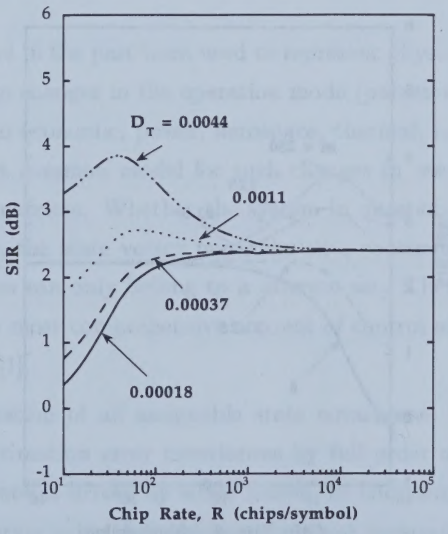


Figure 4. Required SIR for $\overline{P}_b=10^{-2}$ for Channel A, estimate delay of 62.5, and L = 3.

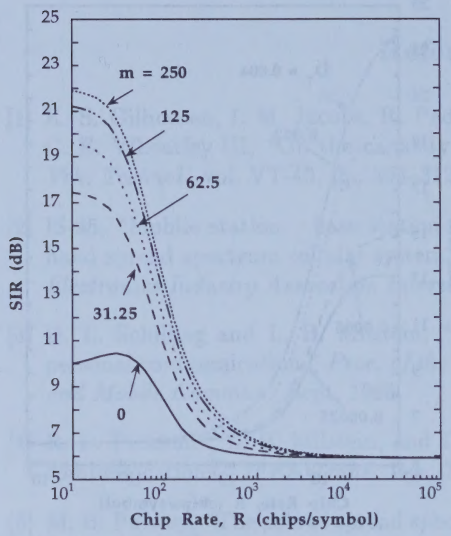


Figure 5. Required SIR for $\overline{P}_b=10^{-2}$ for Channel B, Doppler spread of 0.002, and L = 1.

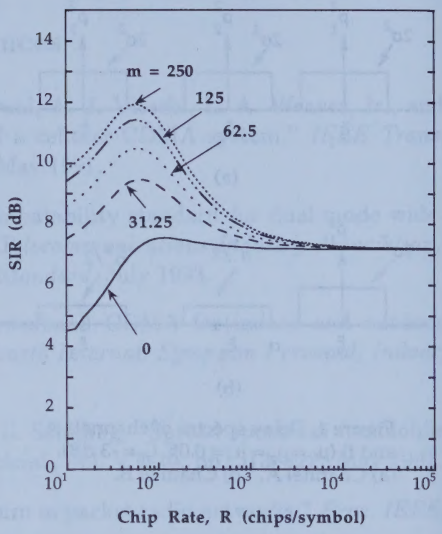


Figure 6. Required SIR for $\overline{P}_b=10^{-2}$ for Channel B, Doppler spread of 0.002, and L = 3.

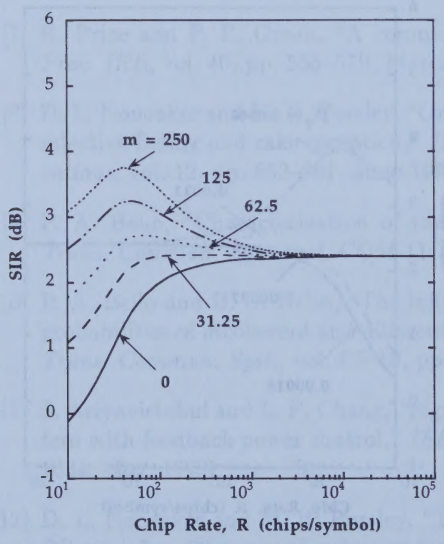


Figure 7. Required SIR for $\overline{P}_b=10^{-2}$ for Channel A, Doppler spread of 0.0011, and L = 3.

PARAMETRIZATION OF ALL MEAN-SQUARE STABLE CONTROLLERS AND OBSERVERS FOR DISCRETE-TIME STOCHASTIC HYBRID SYSTEMS

Edwin Engin Yaz
Electrical Engineering Department
University of Arkansas
Fayetteville, AR 72701

Abstract.

Discrete stochastic hybrid models are considered with Markov jump parameter changes and additive white noises. In both full state feedback control and full order observation problems, characterizations of all assignable steady-state covariances that guarantee mean-square stability and parametrization of gains that assign them are presented. Regional eigenvalue assignment properties of both controllers and observers are also given.

INTRODUCTION

Stochastic hybrid system models have in the past been used to represent physical phenomena that involve sudden and large changes in the operation mode (parameter matrices) of systems with applications to economic, power, aerospace, thermal, and manufacturing systems [1]-[7]. The most common model for such changes in use is the Markov jump model for parameter matrices. Whether the system in question is operating in continuous or discrete time, the state vector may take on a continuum of values whereas the parameter matrices can only belong to a discrete set. Therefore, such models are called hybrid. The most comprehensive account of control and estimation of hybrid systems is given in [1].

The present work is on characterization of all assignable state covariances by full state feedback and all assignable estimation error covariances by full order observers for linear discrete time hybrid systems driven by white noises. In addition, a parametrization of all controller and observer gains that assign a particular covariance is given together with resulting regional eigenvalue assignment properties. Covariance assignability is a necessary and sufficient condition for the mean-square stability of

the closed loop system in control and the mean square convergence of the estimation error in estimation problems. Hence, we indirectly give a parametrization of all mean-square stable controllers and observers for such system models. Only the gains dependent on the knowledge of the current operating mode will be found. This will provide the basis for the solution of the more difficult mode-independent problems. Continuous time version of these results are reported in [8].

CONTROL PROBLEM

Let us consider the discrete time stochastic hybrid system

$$x_{k+1} = A(r_k)x_k + B(r_k)u_k + D(r_k)v_k \tag{1}$$

where $x \in R^n$ is the state, $u \in R^m$ the control input, and $v \in R^n$ the standard white noise (time-wise independent, with zero mean, and unit covariance). The parameter matrices A, B, and D are dependent on a homogeneous Markov chain (which is independent of v_k for all $k \geq 0$), with a finite (N) set of values with transition probabilities

$$\pi_{ij} = prob(r_{k+1} = j | r_k = i) \tag{2}$$

where $prob(r_0 = i) = \pi_i$, $\sum_{j=1}^{q} \pi_{ij} = 1$. x_0 will be independent of all the other noise sources. When $r_k = i$, the parameter matrices will be denoted as

$$(A(r_k), B(r_k), D(r_k) = (A_i, B_i, D_i)$$

for $i = 1, ..., N$. To facilitate the ease of discussion, $D_i D_i^T > 0$ is taken.It is assumed that both the current state vector and the operation mode (the state of the Markov chain) are available and problems of the assignability of given steady state second moments (which will be called covariances following the convention in [9]) and derivation of the feedback gains $K(r_k)$ in

$$u_k = K(r_k)x_k = K_i x_k \tag{3}$$

that achieves a particular covariance assignment will be considered. For the closed loop system, the conditional state covariances are defined as

$$X_i = \lim_{k \to \infty} E\{x_k x_k^T | r_k = i\} \tag{4}$$

with $E\{.\}$ denoting expectation, if the steady state exists, and they satisfy

$$X_i = (A_i + B_i K_i) \sum_{j=1}^{N} \pi_{ij} X_j (A_i + B_i K_i)^T + D_i D_i^T \tag{5}$$

The following theorem sums up the results on the assignability of a certain set of conditional covariances X_i, $i = 1, ..., N$. In the proof of the following two theorems, we made use of the related results in [9].

Theorem 1: The set of covariances $\{X_i > 0, i = 1, ..., N\}$ defined by (4) and satisfying (5) are assignable by the gains $\{K_i,\ i = 1, ..., N\}$ if and only if

$$X_i \geq D_i D_i^T > 0 \tag{6a}$$

and

$$(I - B_i B_i^{\dagger})(X_i - A_i \sum_{j=1}^{N} \pi_{ij} X_j A_i^T - D_i D_i^T)(I - B_i B_i^{\dagger}) = 0 \tag{6b}$$

with $(.)^{\dagger}$ denoting the Moore-Penrose generalized inverse.

Due to the concise nature of this exposition, the proofs of the results reported herein will not be given.

The following result is presented on the feedback gains that achieve a given covariance assignment:

Theorem 2: An assignable set of covariances $\{X_i > 0, i = 1, ..., N\}$ can be assigned by the non-unique set of feedback gains

$$K_i = B_i^{\dagger}(L_i U_i T_i^{-1} - A_i) + (I - B_i^{\dagger} B_i) Z_i \tag{7}$$

where Z_i is arbitrary and

$$L_i L_i^T = X_i - D_i D_i^T,\ T_i T_i^T = \sum_{j=1}^{N} \pi_{ij} X_j > 0 \tag{8}$$

$$U_i = V_{2i}\text{block} - \text{diag}[\text{I}, \bar{\text{U}}_i]\text{V}_{3i}{}^{\text{T}} \tag{9}$$

with V_{2i} and V_{3i} coming from the singular value decompositions

$$(I - B_i B_i^{\dagger}) L_i = V_{1i} \Sigma_i V_{2i}{}^T \tag{10}$$

$$(I - B_i B_i^{\dagger}) A_i T_i = V_{1i} \Sigma_i V_{3i}^T \tag{11}$$

the dimension of the identity in (9) being equal to the rank of Σ_i and $\bar{U}_i$ being arbitrary orthogonal.

Note that when we have a single input system, the last term in (7) is zero. Note also that the assignability conditions of X_i given by Theorem 1 are equivalent to the mean square stabilizability of the closed loop system for $D_i D_i^T > 0$ according to [10], so the results in Theorem 1 can be interpreted as the mean square stabilizability conditions for this kind of systems. By the same token, Theorem 2 can be viewed as the parametrization of all mean square stabilizing state feedback controllers for these models. Mean square stability is also sufficient for the almost sure stability for this class of systems [1].

Next, the regional pole-placement properties of these controllers are considered:

Theorem 3: Let the assignability conditions of Theorem 1 hold. Then the controllers given by Theorem 2 will assign all the eigenvalues $\lambda^i = \lambda(A_i + B_i K_i)$ of the closed loop system for operation modes $i = 1, ..., N$ inside a torus described by

$$\lambda_{max}[(X_i - D_i D_i^T)(\sum_{j=1}^{N} \pi_{ij} X_j)^{-1}] \geq |\lambda^i|^2 \geq \lambda_{min}[(X_i - D_i D_i^T)(\sum_{j=1}^{N} \pi_{ij} X_j)^{-1}] \tag{12}$$

where $\lambda_{max}[.]$ and $\lambda_{min}[.]$ respectively denote the largest and the smallest eigenvalues of the corresponding matrix $[.]$.

The results of [11] are used to prove the above theorem.

OBSERVER PROBLEM

Since there is no duality between the control and estimation problems in covariance assignment, the observer problem is considered separately. Consider system (1) together with the measurement equation

$$y_k = C(r_k) x_k + F(r_k) w_k \tag{13}$$

where $y_k \in R^p$ provides the partial state information, C and F are dependent on the same Markov chain, and w_k is standard white noise independent of x_k, r_k and v_k for all $k \geq 0$. $F_i F_i^T > 0$, $i = 1, ..., N$ will be assumed to simplify the discussion. The following linear unbiased observer which is the discrete counterpart of that given in [1] is used

$$\hat{x}_{k+1} = A_i \hat{x}_k + B_i u_k + G_i (y_k - C_i \hat{x}_k) \tag{14}$$

for reconstruction of state estimates based on the measurements (13) and the current mode information $r_k = i$. The use of this observer yields the estimation error e_k equation

$$e_{k+1} = (A_i - G_i C_i)e_k + D_i v_k - G_i F_i w_k \tag{15}$$

for the model (1) and (13). The steady state conditional estimation error covariance set is defined as

$$P_i = \lim_{k\to\infty} E\{e_k e_k^T | r_k = i\} \tag{16}$$

if it exists, in which case, it satisfies

$$P_i = (A_i - G_i C_i)\sum_{j=1}^{N} \pi_{ij} P_j (A_i - G_i C_i)^T + D_i D_i^T + G_i F_i F_i^T G_i^T \tag{17}$$

The following result is on the assignability of a particular set of covariances. The results of [12] are used in the proofs of the two theorems below.

Theorem 4: A given set of steady state conditional estimation error covariances $\{P_i > 0,\ i = 1, ..., N\}$ are assignable if and only if there exist $L_i{}^o \in R^{n\times p},\ i = 1, ..., N$, such that

$$\begin{aligned} P_i = A_i \sum_{j=1}^{N} \pi_{ij} P_j A_i^T - A_i \sum_{j=1}^{N} \pi_{ij} P_j C_i^T (C_i \sum_{j=1}^{N} \pi_{ij} P_j C_i^T + F_i F_i^T)^{-1} \\ \times C_i \sum_{j=1}^{N} \pi_{ij} P_j A_i^T + D_i D_i^T + L_i{}^o L_i{}^{oT} \end{aligned} \tag{18}$$

The minimal covariance set in the sense of $P_i{}^o \le P_i$ for all P_i satisfying (18) is given for $L_i{}^o = 0$ for all $i = 1, ..., N$.

We also have the following result on the observer gains necessary for assignment:

Theorem 5: A given set of assignable estimation error covariances $\{P_i > 0,\ i = 1, ..., N\}$ can be assigned for the observer (14) by the following set of gains

$$G_i = L_i{}^o U_i{}^o T_i^{o-1} + A_i \sum_{j=1}^{N} \pi_{ij} P_j C_i^T (C_i \sum_{j=1}^{N} \pi_{ij} P_j C_i^T + F_i F_i^T)^{-1} \tag{19}$$

where $L_i{}^o$ is defined in (18), $U_i{}^o \in R^{p\times p}$ arbitrary orthogonal, and

$$T_i^o T_i^{oT} = C_i \sum_{j=1}^{N} \pi_{ij} P_j C_i^T + F_i F_i^T > 0 \tag{20}$$

The optimal gain G_i^o that assigns P_i^o is given by (19) with $L_i^o = 0$, for $i = 1, ..., N$.

Note that due to our earlier assumption $D_i D_i^T > 0$, the existence of a unique set $\{P_i > 0,\ i = 1, ..., N\}$ satisfying (17) or equivalently (18) is necessary and sufficient for the mean square convergence of the estimation error equation (15) according to [10]. So, Theorem 4 can be viewed as providing necessary and sufficient conditions for the mean square detectability of model (1), (13). Also, when we have a single output system, the only possible choices for V_i^o are ± 1. Finally, we present the corresponding results on regional eigenvalue assignment:

Theorem 6: If the assignability conditions of Theorem 4 are satisfied, then the observer (14) with a gain sequence given by (19)-(20) will have its eigenvalues $\lambda^i = \lambda(A_i - G_i C_i)$ for operation modes $i = 1, ..., N$ located in the torus

$$\lambda_{max}[(P_i - D_i D_i^T - G_i F_i F_i^T G_i^T)(\sum_{j=1}^{N} \pi_{ij} P_j)^{-1}] \geq |\lambda^i|^2$$

$$\geq \lambda_{min}[(P_i - D_i D_i^T - G_i F_i F_i^T G_i^T)(\sum_{j=1}^{N} \pi_{ij} P_j)^{-1}] \tag{21}$$

DISCUSSION

Discrete time systems with Markov jump parameters and additive white noise inputs have been considered. Mean square stabilizability by mode-dependent state feedback and mean square detectability by full- order observers using mode-dependent gains have been characterized in terms of assignability of corresponding covariance matrices. Parametrization of all mean square stabilizing controllers and observers have been given using these covariance matrices. It has been shown that once the assignability conditions are satisfied, using these gains in control or estimation problems leads to a specific type of regional eigenvalue placement for each operation mode.

Although it is very easy to test the assignability of a given set of covariances by using Theorem 1 or 3, it would be a lot more difficult to generate assignable covariances for a particular model if we do not have a prior choice for a covariance to be assigned. Even a more difficult problem is the characterization of all covariances assignable by non-switching (operation mode-independent) gains and the parametrization of all mean square stabilizing controllers and observers using non-switching gains. The positive results of our research in this direction are hoped to be available soon.

REFERENCES

1. M.Mariton, Jump Linear Systems in Automatic Control, Marcel Dekker, New York, N.Y., 1990.

2. W.P.Blair and D.D.Sworder, "Feedback control of a class of linear discrete time systems with jump parameters and quadratic cost criteria," Int.J.Contr., v.21, 833-841, 1975.

3. A.S.Willsky, "A survey of design methods for failure detection in dynamic systems," Automatica, v.12, 601-611, 1976.

4. D.D.Sworder and R.O.Rogers, "An LQ solution to a control problem associated with a solar thermal control receiver," IEEE Trans. Autom. Contr., v.28, 971-978, 1983.

5. D.P.Looze, J.L.Weiss, J.Eterno, and N.M.Barrett, "An automatic redesign approach for reconstructable control systems," IEEE Contr. Syst. Mag., v.5, 16-22, 1985.

6. D.D.Sworder and D.S.Chou, "A survey of some design methods for random parameter systems," Proc. 24th CDC, 894-899, Ft.Lauderdale, FL, 1985.

7. R.Akella and P.R.Kumar, "Optimal control of production rate in a failure prone manufacturing system," IEEE Trans. Autom. Contr., v.31, 116, 1986.

8. E.Yaz and I.Yaz, "Parametrization of all mean square stable controllers and observers for stochastic hybrid systems," Proc. 32nd CDC, 863-864, San Antonio, TX, 1993.

9. E.Yaz and R.E.Skelton, "On covariance assignment for linear discrete time systems," Proc. 29th Allerton Conf., 717-726, Monticello, IL, 1991.

10. E.Yaz, "A generalization of the uncertainty threshold principle," IEEE Trans. Autom. Contr., v.35, 942-944, 1990.

11. E.Yaz, R.E.Skelton, and K.Grigoriadis, "Robust regional pole assignment with output feedback, "Proc. 32nd CDC, 3009-3013, San Antonio, TX, 1993.

12. E.Yaz and R.E.Skelton, "Continuous and discrete state estimation with error covariance assignment," Proc. 30th CDC, 3091-3092, Brighton, U.K., 1991.

Stochastic Approximations for State-Independent Denumerable Markov Chains

Younseok Choo
Quality Assurance Section, ETRI
161 Kajeong-dong Yusong-gu, Taejon, Korea 305-350
Aristotle Arapostathis
Department of Electrical and Computer Engineering
The University of Texas at Austin
Austin, TX 78712-1084, USA

Abstract

In this paper we analyze the convergence of stochastic approximations driven by state-independent denumerable Markov chains via the well-known Ordinary Differential Equation (ODE) method. When the forcing term h is bounded, the differential equation limit of the state process is identified under a recurrence condition on the Markov chain. Using the standard ergodic theory for Markov chains, similar result is obtained for the case where h is unbounded.

1 Introduction

Let Θ be a compact subset of $\Re^r$ and Ξ a denumerable set. We are concerned with the asymptotic behavior of the sequence $\{\theta_n\}$ generated by a recursive algorithm

$$\theta_{n+1} = \theta_n + a_n h(\theta_n, \xi_{n+1}) \tag{1.1}$$

where $\theta_n \in \Omega$, $\xi_n \in \Xi$, h is a $\Re^r$-valued measurable function and $\{a_n\}$ is a sequence of positive real numbers. Since the pioneering work of Robbins and Monro [9], many techniques have been developed to treat the convergence problems related to (1.1). The ODE method [3], [6] is one of the most powerful tools in dealing with those problems. In the ODE method we first associate a differential equation with (1.1), and then analyze the asymptotic behavior of $\{\theta_n\}$ through the stability analysis of the corresponding differential equation.

*This research was supported in part by the Texas Advanced Research Program (Advanced Technology Program) under Grant No. 003658-186, in part by the Air Force Office of Scientific Research under Grant F49620-92-J-0045 and in part by the Post Doc. Progrm of ETRI, Korea.

The successful applications of the ODE method include the Markovian problems where the process $\{\theta_n, \xi_n\}$ in (1.1) is Markovian (e.g., [4], [5], [7], [8]). In this paper we restrict our attention to a particular situation, i.e., the *noise* process $\{\xi_n\}$ is independent of the *state* process and forms a Markov chain on the denumerable set Ξ. When the forcing term h is bounded, the differential equation limit of the state process is identified under a recurrence condition on the Markov chain. Using the standard ergodic theory for Markov chains, similar result is obtained for the case where h is unbounded.

Throughout this paper an underlying probability space $(\Omega, \mathcal{F}, \mathcal{P})$ is assumed to be given so that $\{\theta_n\}$ and $\{\xi_n\}$ in (1.1) are sequences of random variables defined on it.

Let $P := [p_{ij}]$ denote the probability transition matrix for the Markov chain $\{\xi_n\}$ and, for each $m \geq 1$, $P^m := [p_{ij}^m]$ the m-fold product of P with itself. Let T_{ij} denote the mean first entrance time from i to j and C_{ij} the expected *cost* during a first passage from i to j, i.e., if $\tau_j = \inf\{n \geq 1 : \xi_n = j\}$, then

$$T_{ij} = E[\tau_j \mid \xi_0 = i], \tag{1.2}$$

$$C_{ij}(\theta) = E[\sum_{k=0}^{\tau_j - 1} h(\theta, \xi_k) \mid \xi_0 = i]. \tag{1.3}$$

For each $\theta \in \Theta$ consider the Poisson equation

$$v(\theta, \cdot) + \overline{h}(\theta) = h(\theta, \cdot) + Pv(\theta, \cdot) \tag{1.4}$$

where $Pv(\theta, i) = \sum_{j \in \Xi} p_{ij} v(\theta, j)$, $i \in \Xi$. A pair $(\overline{h}(\theta), v(\theta, \cdot))$ satisfying (1.4) is a solution to the Poisson equation.

For a moment assume that there is a state, say $0 \in \Xi$, such that $T_{i0} < \infty$ for all $i \in \Xi$. Then a solution pair $(\overline{h}(\theta), v(\theta, \cdot))$ to the Poisson equation is given by [2], [10]

$$\overline{h}(\theta) = C_{00}(\theta)/T_{00}, \tag{1.5}$$

$$v(\theta, i) = C_{i0}(\theta) - \overline{h}(\theta) T_{i0}, \ i \in \Xi. \tag{1.6}$$

If the Markov chain $\{\xi_n\}$ has a unique invariant probability $\pi(\cdot) = [\pi(i) : i \in \Xi]$ and v is bounded, then $\overline{h}(\theta)$ in (1.5) can be expressed by

$$\overline{h}(\theta) = \sum_{j \in \Xi} h(\theta, j) \pi(j). \tag{1.7}$$

To see this, fix $i \in \Xi$. Then from the Poisson equation

$$v(\theta, i) + \overline{h}(\theta) = h(\theta, i) + \sum_{j \in \Xi} p_{ij} v(\theta, j). \tag{1.8}$$

Multiplying (1.8) by $\pi(i)$ and taking summation over i, we have

$$\begin{aligned} \sum_{i \in \Xi} \pi(i) v(\theta, i) + \overline{h}(\theta) &= \sum_{i \in \Xi} \pi(i) h(\theta, i) + \sum_{i \in \Xi} \sum_{j \in \Xi} \pi(i) p_{ij} v(\theta, j) \\ &= \sum_{i \in \Xi} \pi(i) h(\theta, i) + \sum_{j \in \Xi} \pi(j) v(\theta, j). \end{aligned}$$

2 Main Results

First we deal with a case where h is bounded. The assumptions (A1)-(A4) below are imposed on the algorithm (1.1).

(A1) $a_n \downarrow 0$, $\sum_{n=0}^{\infty} a_n = \infty$ and $\sum_{n=0}^{\infty} a_n^2 < \infty$.
(A2) h is bounded on $\Theta \times \Xi$. The family of functions $\{h(\cdot, \xi) : \xi \in \Xi\}$ is equicontinuous.
(A3) The Markov chain $\{\xi_n\}$ has a unique invariant probability $\pi(\cdot) = [\pi(i) : i \in \Xi]$.
(A4) There exist a state, say $0 \in \Xi$, and a finite number K such that

$$T_{i0} \leq K \quad \text{for all } i \in \Xi.$$

Let $t_n = \sum_{i=0}^{n-1} a_i$ and let $\theta^0(\cdot)$ denotes a piecewise linear interpolation function with interpolation steps $\{a_n\}$ such that $\theta^0(t_n) = \theta_n$, and $\theta^0(t) = \theta_0$ for $t < 0$. $\{\theta^n(\cdot)\}$ is a sequence of left shifts of $\theta^0(\cdot)$ defined by $\theta^n(t) = \theta^0(t + t_n)$. Our first result is stated in the following theorem.

Theorem 1 *For the algorithm defined by (1.1), assume (A1)-(A4) hold. Then each sequence in $\{\theta^n(\cdot)\}$ has a convergent subsequence and the limit $\theta(\cdot)$ of any convergent subsequence is a solution to the differential equation*

$$\dot{\theta}(t) = \overline{h}(\theta(t)) \tag{2.1}$$

on $(-\infty, \infty)$ a.s., where $\overline{h}$ is as defined in (1.7).

Proof: Rewrite (1.1) as

$$\begin{aligned} \theta_{n+1} &= \theta_n + a_n \overline{h}(\theta_n) + a_n(h(\theta_n, \xi_{n+1}) - \overline{h}(\theta_n)) \\ &:= \theta_n + a_n \overline{h}(\theta_n) + a_n \epsilon_n. \end{aligned} \tag{2.2}$$

According to [8] (see [3] also), it is sufficient to show that for each $T > 0$

$$\lim_{n \to \infty} \sup_{n \leq k \leq m(n,T)} |\sum_{i=n}^{k} a_i \epsilon_i| = 0 \quad \text{a.s.} \tag{2.3}$$

where $m(n,T) = \min\{k > n : \sum_{i=n}^{k} a_i \leq T\}$. ¿From the Poisson equation $\epsilon_n = v(\theta_n, \xi_{n+1}) - Pv(\theta_n, \xi_{n+1})$. We decompose ϵ_n as $\epsilon_n = x_n + y_n + z_n$, where

$$\begin{aligned} x_n &= v(\theta_n, \xi_{n+1}) - Pv(\theta_n, \xi_n), \\ y_n &= Pv(\theta_n, \xi_n) - Pv(\theta_{n+1}, \xi_{n+1}), \\ z_n &= Pv(\theta_{n+1}, \xi_{n+1}) - Pv(\theta_n, \xi_{n+1}). \end{aligned}$$

It is shown in [7], [8] that $\{a_n x_n\}$ and $\{a_n y_n\}$ satisfy (2.3). To prove the same thing for $\{a_n z_n\}$, we show the family of functions $\{v(\cdot, \xi) : \xi \in \Xi\}$ is equicontinuous. Then we have $z_n \to 0$ since $|\theta_{n+1} - \theta_n| \to 0$, by which $\{a_n z_n\}$ satisfies (2.3).

Note that from (1.3)

$$C_{i0}(\theta) = \sum_{n=1}^{\infty} \mathcal{P}[\tau_0 = n \mid \xi_0 = i] \sum_{m=0}^{n-1} \sum_{j \in \Xi} p_{ij}^m h(\theta, j).$$

Fix $\epsilon > 0$. Then, by (A2), there exists $\delta > 0$ such that for each $i \in \Xi$

$$| h(\theta_1, i) - h(\theta_2, i) | \leq \epsilon / K \quad \text{whenever } | \theta_1 - \theta_2 | \leq \delta.$$

If $| \theta_1 - \theta_2 | \leq \delta$, then

$$\begin{aligned} & | C_{i0}(\theta_1) - C_{i0}(\theta_2) | \\ \leq & \sum_{n=1}^{\infty} \mathcal{P}[\tau_0 = n \mid \xi_0 = i] \sum_{m=0}^{n-1} \sum_{j \in \Xi} p_{ij}^m | h(\theta_1, j) - h(\theta_2, j) | \\ \leq & (\epsilon / K) \sum_{n=1}^{\infty} n \mathcal{P}[\tau_0 = n \mid \xi_0 = i] \\ = & (\epsilon / K) T_{i0} \\ \leq & \epsilon \end{aligned}$$

and so $\{C_{i0}(\cdot)\}$ is equicontinuous, which in turn implies the equicontinuity of $\{v(\cdot, i)\}$. **Q.E.D.**

Now we consider a case where h is unbounded. Consider the following assumptions.

(B1) $\sum_{n=0}^{\infty} a_n = \infty$ and $\lim_{n \to \infty} n a_n = b$ for some $0 < b < \infty$.

(B2) The family of functions $\{h(\cdot, \xi) : \xi \in \Xi\}$ is equi-Lipschitzian, i.e., there exists $L > 0$ such that for all $\xi \in \Xi$

$$| h(\theta, \xi) - h(\theta', \xi) | \leq L | \theta - \theta' | .$$

(B3) The Markov chain $\{\xi_n\}$ is aperiodic, irreducible and positive recurrent (hence $\{\xi_n\}$ has a unique invariant probability, denoted by $\pi(\cdot) = [\pi(i) : i \in \Xi]$).

(B4) There exist $C > 0$ and a positive real-valued Borel-measurable function g defined on Ξ such that

(i) $\sup_\theta | h(\theta, \xi) | \leq C(1 + g(\xi))$,

(ii) $\overline{g} := \sum_{\xi \in \Xi} g(\xi) \pi(\xi) < \infty$.

Under (B3) and (B4), the standard ergodic theory for Markov chains [1] implies that there exists a null set Ω_o (independent of θ) such that for all $\omega \notin \Omega_o$, $\theta \in \Theta$

$$\begin{aligned} \lim_{n \to \infty} \frac{1}{n+1} \sum_{i=0}^{n} h(\theta, \xi_i) &= \sum_{j \in \Xi} h(\theta, j) \pi(j) \\ &:= \overline{h}(\theta). \end{aligned} \tag{2.4}$$

Fix $\theta \in \Theta$ and let $\alpha > 0$. Then, by (B1), (2.4) and [8, Lemma D], there is $N(\alpha, \theta)$ such that for all $m > n > N(\alpha, \theta)$

$$| \sum_{i=n}^{m} a_i (h(\theta, \xi) - \overline{h}(\theta)) | \leq \alpha (1 + \sum_{i=n}^{m} a_i).$$

Therefore we have that for each $T > 0$ and $\theta \in \Theta$

$$\lim_{n\to\infty} \sup_{n\le k\le m(n,T)} | \sum_{i=n}^{k} a_i(h(\theta, \xi_{i+1}) - \overline{h}(\theta)) | = 0. \tag{2.5}$$

Actually the convergence in (2.5) is uniform in θ as can be shown below.

Lemma 1 *For each $T > 0$ the convergence in (2.5) is uniform in θ a.s.*

Proof: Fix $\omega \notin \Omega_o$, $T > 0$ and $\epsilon > 0$. Let δ be such that $0 < \delta \le \frac{\epsilon}{4LT}$. Let $B_\delta(\theta)$ denote the open ball with radius δ centered at θ. Since Θ is compact, there exists a finite set $\{\theta^l : 1 \le l \le p\}$ in Θ such that

$$\Theta \subseteq \bigcup_{l=1}^{p} B_\delta(\theta^l).$$

For each θ^l, $1 \le l \le p$, there exists N_l such that for all $n \ge N_l$

$$\sup_{n\le k\le m(n,T)} | \sum_{i=n}^{k} a_i(h(\theta^l, \xi_{i+1}) - \overline{h}(\theta^l)) | \le \epsilon/2.$$

Let $N_\epsilon = \max\{N_l : 1 \le l \le p\}$. Choose $\theta \in \Theta$ arbitrarily. Then $\theta \in B_\delta(\theta^l)$ for some l. If $n \ge N_\epsilon$, then

$$\begin{aligned}
\sup_{n\le k\le m(n,T)} | \sum_{i=n}^{k} a_i(h(\theta, \xi_{i+1}) - \overline{h}(\theta)) | \;&\le\; \sup_{n\le k\le m(n,T)} | \sum_{i=n}^{k} a_i(h(\theta, \xi_{i+1}) - h(\theta^l, \xi_{i+1})) | \\
&+ \sup_{n\le k\le m(n,T)} | \sum_{i=n}^{k} a_i(h(\theta^l, \xi_{i+1}) - \overline{h}(\theta^l)) | \\
&+ \sup_{n\le k\le m(n,T)} | \sum_{i=n}^{k} a_i(\overline{h}(\theta^l) - \overline{h}(\theta)) | \\
&\le LT \mid \theta - \theta^l \mid + \epsilon/2 + LT \mid \theta - \theta^l \mid \\
&\le \epsilon.
\end{aligned}$$

Q.E.D.

Now we are ready to state our second result.

Theorem 2 *Under (B1)-(B4), the conclusions of Theorem 1 continue to hold.*

Proof: Rewrite (1.1) as

$$\theta_{n+1} = \theta_n + a_n \overline{h}(\theta_n) + a_n \epsilon_n$$

where $\epsilon_n = h(\theta_n, \xi_{n+1}) - \overline{h}(\theta_n)$. As in the proof of Theorem 1, it is sufficient to show that the sequence $\{a_n \epsilon_n\}$ satisfies (2.3). Consider the decomposition

$$\sum_{i=n}^{k} a_i \epsilon_i \;=\; \sum_{i=n}^{k} a_i(h(\theta_i, \xi_{i+1}) - h(\theta_n, \xi_{i+1}))$$

$$
\begin{aligned}
&+ \sum_{i=n}^{k} a_i(\overline{h}(\theta_n) - \overline{h}(\theta_i)) \\
&+ \sum_{i=n}^{k} a_i(h(\theta_n, \xi_{i+1}) - \overline{h}(\theta_n)) \\
&:= B_1(n,k) + B_2(n,k) + B_3(n,k).
\end{aligned}
$$

Fix $T > 0$. Lemma 1 shows that

$$
\lim_{n\to\infty} \sup_{n\le k\le m(n,T)} \mid B_3(n,k) \mid = 0.
$$

Therefore it is sufficient to show that for $j = 1, 2$

$$
\lim_{n\to\infty} \sup_{n\le k\le m(n,T)} \mid B_j(n,k) \mid = 0. \tag{2.6}
$$

We proceed as in [8]. Fix $a \in (0,1)$ and let $\tau = \sqrt{a}T$. Define for each n

$$
n = k_0 < k_1 < \cdots < k_r = m(k_{r-1}, \tau) \cdots .
$$

Then for each r, $k_r \le k < k_{r+1}$

$$
\begin{aligned}
\mid B_1(k_r,k) \mid &\le L \sum_{i=k_r}^{k} a_i \mid \theta_i - \theta_{k_r} \mid \\
&\le L \sum_{i=k_r}^{k} a_i \sum_{j=k_r}^{i-1} a_j(1 + g(\xi_{j+1}))C.
\end{aligned}
$$

Since (2.4) holds for $g(\cdot)$, there exists $N(a)$ such that for all $m \ge n \ge N(a)$

$$
\mid \sum_{i=n}^{m} a_i[C(1 + g(\xi_{i+1})) - \overline{g}] \mid \le a(1 + \sum_{i=n}^{m} a_i).
$$

Assume $n \ge N(a)$. Then for each r, $k_r \le k \le k_{r+1}$

$$
\begin{aligned}
\mid B_1(k_r,k) \mid &\le L \sum_{i=k_r}^{k} a_i[a(1 + \sum_{j=k_r}^{i-1} a_j) + \overline{g} \sum_{j=k_r}^{i-1} a_j] \\
&\le L \sum_{i=k_r}^{k} a_i[a(1+\tau) + \tau\overline{g}] \\
&\le L[a(1+\tau) + \tau\overline{g}]\tau \\
&\le \overline{L}a
\end{aligned}
$$

for some $\overline{L} > 0$. Similarly for each r, $k_r \le k < k_{r+1}$

$$
\mid B_2(k_r,k) \mid \le \overline{L}a.
$$

Since $t_{k_r} - t_{k_{r-1}} \ge \tau/2$ for sufficiently large n,

$$
t_{k_r} \ge t_{m(n,T)} \quad \text{if } r \ge 2/\sqrt{a}.
$$

Then for each $N(a) \le n \le k \le m(n,T)$, $j = 1, 2$

$$
\mid B_j(n,k) \mid \le 2\overline{L}(a + \sqrt{a})
$$

which implies (2.6). **Q.E.D.**

References

[1] K.L. Chung, *Markov chains with stationary transition probabilities*, Springer-Verlag, N.Y., 1967.

[2] C. Derman and A. F. Veinott, "A solution to a countable system of equations arising on Markovian decision processes," *Ann. Math. Statist.*, vol. 38, pp. 582-584, 1967.

[3] H.J. Kushner and D.S. Clark, *Stochastic approximation methods for constrained and unconstrained systems*,Springer-Verlag, N.Y., 1978.

[4] H.J. Kushner and A. Shwartz, "An invariant measure approach to the convergence of stochastic approximations with state dependent noise," *SIAM J. Contr. Optim.*, vol. 22, pp. 13-27, 1984.

[5] H.J. Kushner, "Stochastic approximation with discontinuous dynamics and state dependent noise: w.p. 1 and weak convergence," *J. Math. Anal. Appl.*, vol. 82, pp. 527-542, 1981.

[6] L. Ljung, "Analysis of recursive stochastic algorithms," *IEEE Trans. Automat. Contr.*, vol. 22, pp. 551-575, 1977.

[7] D.-J. Ma, A. M. Makowski and A. Shwartz, "Stochastic approximations for finite-state Markov chains," *Stoch. Proc. Appl.*, vol. 35, pp. 27-45, 1990.

[8] M. Metivier and P. Priouret, "Applications of a Kushner and Clark lemma to general classes of stochastic algorithms," *IEEE Trans. Inform. Th.*, vol. 30, pp. 140-151, 1984.

[9] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. of Math. Stat.*, vol. 22, pp. 400-407, 1951.

[10] A. Shwartz And M. Makowski, *On the Poisson equation for Markov chains*, Report No. EE-646, Faculty of Electrical Engineering, Technion: Israel Institute of Technology, 1987.

Necessity of Kushner–Clark Condition for Convergence of Stochastic Approximation Algorithms

I-Jeng Wang and Edwin K. P. Chong
School of Electrical Engineering
Purdue University, West Lafayette, IN 47907
iwang, echong@ecn.purdue.edu

Sanjeev R. Kulkarni
Department of Electrical Engineering
Princeton University, Princeton, NJ 08544
kulkarni@ee.princeton.edu

Abstract

In this paper, we study the convergence of Stochastic Approximation algorithms. We establish the necessity of the Kushner–Clark condition for convergence of algorithms by proving its equivalence with a new necessary and sufficient condition proposed by Kulkarni and Horn. The result suggests that the popular sufficient condition cannot be further relaxed. We also present a simplified equivalence of the Kushner–Clark condition.

1 Introduction

The *Stochastic Approximation* (SA) algorithm has been a popular technique for solving stochastic local optimization problems where estimates or noisy observations of the gradients are available, since first introduced by Robbins and Monro [1]. Studies of the convergence of SA algorithms constitutes a major part of the research in this field. Various conditions for the convergence of SA algorithms have been proposed over the years, for examples [1, 2, 3, 4, 5]. One of the most important sufficient conditions is provided by the Kushner–Clark theorem [2, 3]. Since this condition has been adopted in numerous applications, it is natural to question whether the sufficient condition is too strong. We answer the above question partially by giving a necessary and sufficient version of the Kushner–Clark theorem for the convergence of a class of Robbins–Monro algorithms. We accomplish this by establishing the equivalence between the Kushner–Clark condition and a new necessary and sufficient condition proposed by Kulkarni and Horn [5]. Our result shows that, under appropriate assumptions, the Kushner–Clark condition is

This work was supported in part by a Purdue Research Foundation Fellowship, by the Engineering Foundation through grant RI-A-93-09, and by a Whirlpool Faculty Fellowship.

necessary for convergence. In this case, we conclude that the Kushner–Clark condition cannot be further relaxed. We also prove that the Kushner–Clark condition can be simplified.

In the following section we state briefly both conditions and slightly extend the Kulkarni and Horn's result to a more general class of algorithms. We prove the equivalence of the two conditions and state the necessary and sufficient version of the Kushner–Clark theorem in section 3. We also give a simplified equivalence of the Kushner–Clark condition. Finally we conclude in section 4.

2 Conditions for Convergence of SA Algorithms

One of the most popular and frequently used sufficient conditions for the convergence of SA is the Kushner–Clark theorem [2, 3]. A simplified version of the theorem is given below:

Theorem 1 *[3] Consider the stochastic algorithm*

$$x_{n+1} = x_n - a_n f(x_n) + a_n r_n + a_n \epsilon_n, \tag{1}$$

where x_n, r_n, ϵ_n are real valued random variables, f is a continuous function from $\mathbb{R}$ into $\mathbb{R}$, and $\{a_n\}$ is a positive sequence converging to 0 with $\sum_{n=1}^{\infty} a_n = \infty$. Suppose the following are true:

1. $\sup_n |x_n| < \infty$ *almost surely;*
2. x^* *is the unique asymptotically stable equilibrium of* $\frac{dx(t)}{dt} = -f(x(t))$, $t \in (-\infty, \infty)$;
3. $\lim_n r_n = 0$ *almost surely; and*

(KC) For every $T > 0$,

$$\lim_{n\to\infty} \left(\sup_{n<p\leq m(n,T)} \left| \sum_{i=n+1}^{p} a_i \epsilon_i \right| \right) = 0 \text{ almost surely,}$$

where $m(n,T) \triangleq \max\{k : a_n + \cdots + a_k \leq T\}$.

Then

$$\lim_{n\to\infty} x_n = x^* \text{ almost surely.}$$

We will refer to the condition (KC) as the Kushner–Clark condition on the noise sequence and the limit in (KC) as the KC limit. Throughout the paper, we will assume that the step-size $\{a_n\}$ satisfies the following condition as in the above theorem:

(SC) $a_n > 0, a_n \to 0$ and $\sum_{n=1}^{\infty} a_n = \infty$.

Recently, Kulkarni and Horn [5] established a necessary and sufficient condition for convergence of the Robbins–Monro algorithm. They considered a simple scalar algorithm that can be described by

$$x_{n+1} = x_n - a_n(f(x_n) + e_n), \tag{2}$$

where $\{a_n\}$ satisfies (SC) and $\{e_n\}$ is the noise sequence. Kulkarni and Horn's condition is in terms of the following deterministic property of the noise sequence:

Definition 1 [5] Fix a positive sequence $\{a_n\}$. We say that the noise sequence $\{e_n\}$ is *persistently disturbing* with respect to $\{a_n\}$ if there exist constants $\alpha, \beta > 0$ and an infinite set of nonoverlapping intervals $\{I_k\}$ such that for all k,

$$\left| \sum_{n \in I_k} a_n e_n \right| \geq \alpha \sum_{n \in I_k} a_n + \beta.$$

The following theorem gives a necessary and sufficient condition for the convergence of (2):

Theorem 2 *[5] For any function $f : \mathbb{R} \to \mathbb{R}$ satisfying*

1. *There exists a unique point x^* such that $f(x) > 0$ for $x > x^*$ and $f(x) < 0$ for $x < x^*$;*
2. *$|f(x)| \leq K_f$ for all x;*
3. *For all $\delta > 0$, there exists $h_\delta > 0$ such that $|x - x^*| \geq \delta$ implies $|f(x)| \geq h_\delta$; and*
4. *$f(x)$ is continuous at $x = x^*$;*

and any initial point x_1, x_n converges to x^* for the algorithm described by (2) if and only if the noise sequence $\{e_n\}$ is not persistently disturbing.

In the next section, we show that the Kushner–Clark condition on the noise sequence is equivalent to the non–persistent disturbing property of the sequence. This implies that the Kushner–Clark condition is not only sufficient but also necessary under the same assumptions on the function as stated in Theorem 2. Before pursuing the main result, we present a lemma that shows that the additional term r_n appearing in (1) does not affect Kulkarni and Horn's result.

Lemma 1 *Given a sequence $\{a_n\}$ satisfying condition (SC), suppose*

$$e_n = r_n + \epsilon_n, \qquad \lim_{n \to \infty} r_n = 0.$$

Then $\{e_n\}$ is persistently disturbing with respect to $\{a_n\}$ if and only if $\{\epsilon_n\}$ is persistently disturbing with respect to the same sequence.

Proof: ($\Longrightarrow$) If $\{e_n\}$ is persistently disturbing, then there exist $\alpha, \beta > 0$ and an infinite sequence of nonoverlapping intervals $\{I_k\}$ such that, for all k,

$$\left|\sum_{n\in I_k} a_n e_n\right| \geq \alpha \sum_{n\in I_k} a_n + \beta.$$

Since $r_n \to 0$, there exists N such that $|r_n| \leq \alpha - \delta$ with $0 < \delta < \alpha$, for all $n > N$. Let $\{I'_k\} = \{I \in \{I_k\} : \min(I) > N\}$. Then for all k,

$$\left|\sum_{n\in I'_k} a_n\epsilon_n\right| + \sum_{n\in I'_k} a_n|r_n| \geq \left|\sum_{n\in I'_k} a_n(\epsilon_n + r_n)\right| = \left|\sum_{n\in I'_k} a_n e_n\right| \geq \alpha \sum_{n\in I'_k} a_n + \beta.$$

Hence,

$$\begin{aligned}\left|\sum_{n\in I'_k} a_n\epsilon_n\right| &\geq \alpha \sum_{n\in I'_k} a_n - \sum_{n\in I'_k} a_n|r_n| + \beta \\ &\geq (\alpha - (\alpha - \delta)) \sum_{n\in I'_k} a_n + \beta \\ &= \delta \sum_{n\in I'_k} a_n + \beta.\end{aligned}$$

Thus $\{\epsilon_n\}$ is persistently disturbing.

($\Longleftarrow$) Write $\epsilon_n = e_n - r_n$ and apply the proof of the first part with r_n replaced by $-r_n$.

∎

3 Equivalence of the Conditions

We present a theorem to establish the equivalence between the deterministic version of Kushner–Clark condition (KC) and the non–persistent–disturbing of the noise sequence. To proceed with the proof, we need some notation.

We define recursively a class of operations on $n \in \mathbb{N}$ and $T \in \mathbb{R}^+$

$$\begin{aligned}m^0(n,T) &\triangleq n, \\ m^j(n,T) &\triangleq m(m^{j-1}(n,T),T)\ j \geq 1, j \in \mathbb{N}.\end{aligned}$$

For an interval of natural numbers I and a positive real number T, we write:

$$l(I) \triangleq \min(I),\ u(I) \triangleq \max(I),$$

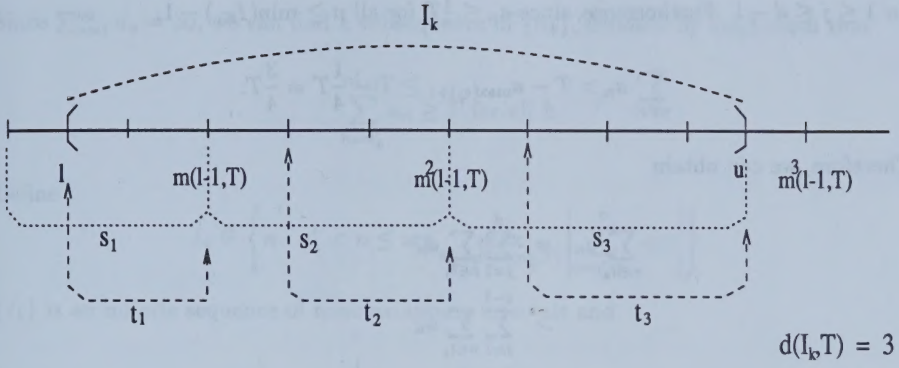


Figure 1: Pictorial description of some notation.

$$
\begin{aligned}
d(I,T) &\triangleq \min\{j : m^j(l-1,T) \geq u\}, \\
s_j(I,T) &\triangleq \{n : m^{j-1}(l-1,T) \leq n \leq \max(m^j(l-1,T),u)\},\ 1 \leq j \leq d, \\
t_j(I,T) &\triangleq s_j(I,T) \setminus \{\min(s_j(I,T))\}.
\end{aligned}
$$

Roughly speaking, $d(I,T)$ is the number of segments in I over which $\{a_n\}$ is summed up to T. Note also that $t_i(I,T) \cap t_j(I,T) = \phi$ for $i \neq j$ and $\cup t_j(I,T) = I$. Later on we shall drop the arguments I and T of the notations defined above when no confusion may occur. Figure 1 gives a pictorial description of this notation. The following lemma is also required to prove the main theorem:

Lemma 2 *Given a sequence $\{a_n\}$ satisfying (SC), an infinite sequence of nonoverlapping intervals $\{I_k\}, I_k \subset \mathbb{N}$, and a positive real number T, there exists a positive integer K_1 such that*

$$\sum_{n \in I_k} a_n > (d(I_k,T) - 1)\frac{T}{2}$$

for all $k \geq K_1$.

Proof: Since $a_n \to 0$, there exists $N \in \mathbb{N}$ such that $a_n \leq \frac{1}{4}T$ for all $n > N$. Choose $K_1 \in \mathbb{N}$ such that

$$\min(I_k) - 1 > N \text{ for all } k \geq K_1.$$

For any I_k with $k \geq K_1$, the desired inequality is trivial if $a_{l-1} + \sum_{n \in I_k} a_n \leq T$ (since $d(I_k,T) - 1 = 0$ in this case) Let us assume that $a_{l-1} + \sum_{n \in I_k} a_n > T$, i.e., $d > 1$. From the definition of s_j and $m^j(n,T)$, we have

$$\sum_{n \in s_j} a_n + a_{\max(s_j)+1} > T,$$

for $1 \leq j \leq d-1$. Furthermore, since $a_n \leq \frac{1}{4}T$ for all $n \geq \min(I_{K_1}) - 1$,

$$\sum_{n \in s_j} a_n > T - a_{\max(s_j)+1} \geq T - \frac{1}{4}T = \frac{3}{4}T.$$

Therefore, we can obtain

$$\begin{aligned}
\sum_{n \in I_k} a_n &= \sum_{j=1}^{d} \sum_{n \in t_j} a_n \\
&> \sum_{j=1}^{d-1} \sum_{n \in t_j} a_n \\
&= \sum_{j=1}^{d-1} \sum_{n \in s_j} a_n - \sum_{j=1}^{d-1} a_{\min(s_j)} \\
&> (d-1)\frac{3}{4}T - (d-1)\frac{1}{4}T \\
&= (d-1)\frac{1}{2}T.
\end{aligned}$$

This completes the proof of the lemma. ∎

We present the main theorem that proves the equivalence between the Kushner–Clark condition and Kulkarni and Horn's condition. The theorem also provides a simpler equivalence of the KC condition.

Theorem 3 *Suppose $\{a_n\}$ satisfies condition (SC). The following three conditions are equivalent:*

1. *The noise sequence $\{e_n\}$ is not persistently disturbing with respect to $\{a_n\}$.*
2. *For every $T > 0$, the KC limit for $\{e_n\}$ converges to zero.*
3. *For some $T > 0$, the KC limit for $\{e_n\}$ converges to zero.*

Proof: $(1 \Longrightarrow 2)$ We shall prove the contrapositive.
Suppose for some $T > 0$,

$$\lim_{n \to \infty} \left(\sup_{n < k \leq m(n,T)} \left| \sum_{i=n+1}^{k} a_i e_i \right| \right) \neq 0.$$

Then there exist $\epsilon > 0$ and an increasing sequence $\{n_k\}, n_k \to \infty$ such that

$$\sup_{n_k < p \leq m(n_k,T)} \left| \sum_{i=n_k+1}^{p} a_i e_i \right| \geq \epsilon \text{ for all } k.$$

Since $\sum_{n=1}^{\infty} a_n = \infty$, we can find a subsequence of $\{n_k\}$, denoted by $\{n'_k\}$, such that

$$\sum_{n=n'_k}^{n'_{k+1}} a_n \geq T \text{ for all } k.$$

Define

$$I_k \triangleq \left\{ n : n'_k < n \leq \arg \max_{n'_k < p \leq m(n'_k, T)} \left| \sum_{i=n'_k+1}^{p} a_i e_i \right| \right\}.$$

$\{I_k\}$ is an infinite sequence of nonoverlapping intervals and

$$\left| \sum_{n \in I_k} a_n e_n \right| \geq \epsilon, \ \sum_{n \in I_k} a_n < T \ \text{ for all } k.$$

Choosing $\beta < \epsilon$ and $\alpha < \frac{\epsilon - \beta}{T}$, we have

$$\begin{aligned} \left| \sum_{n \in I_k} a_n e_n \right| &\geq \epsilon = \frac{\epsilon - \beta}{T} \cdot T + \beta \\ &> \alpha \sum_{n \in I_k} a_n + \beta \ \text{ for all } k. \end{aligned}$$

Therefore $\{e_n\}$ is persistently disturbing with respect to $\{a_n\}$.

$(2 \Longrightarrow 3)$ This part is trivial.

$(3 \Longrightarrow 1)$ Fix $T > 0$, for any $\alpha, \beta > 0$ and an infinite sequence of nonoverlapping intervals $\{I_k\}$, there exists $N_1 \in \mathbb{N}$ such that

$$\sup_{n < p \leq m(n,T)} \left| \sum_{i=n+1}^{p} a_i e_i \right| < \min\left(\frac{T}{2}\alpha, \beta \right) \ \text{ for all } n > N. \tag{3}$$

Choose $K_2 \in \mathbb{N}$ such that $\min(I_k) - 1 > N$ for all $k \geq K_2$. Also by lemma 2, there exists $K_1 \in \mathbb{N}$ such that

$$\sum_{n \in I_k} a_n > (d(I_k, T) - 1)\frac{T}{2} \text{ for all } k \geq K_1. \tag{4}$$

Let $K = \max(K_1, K_2)$. Then for all I_k with $k \geq K$, we consider the following two possibilities:

(i) $a_{l-1} + \sum_{n \in I_k} a_n \leq T$:

By (3) we have

$$\left| \sum_{n \in I_k} a_n e_n \right| \leq \sup_{l-1 < p \leq m(l-1, T)} \left| \sum_{i=l}^{p} a_i e_i \right| < \beta < \alpha \sum_{n \in I_k} a_n + \beta.$$

(ii) $a_{l-1} + \sum_{n \in I_k} a_n > T$:

By (3),

$$\left| \sum_{n \in t_j} a_n e_n \right| < \min(\frac{T}{2}\alpha, \beta), \ 1 \leq j \leq d. \tag{5}$$

From (4), (5) and the triangle inequality, we can obtain

$$\begin{aligned} \left| \sum_{n \in I_k} a_n e_n \right| &\leq \sum_{j=1}^{d-1} \left| \sum_{n \in t_j} a_n e_n \right| + \left| \sum_{n \in t_d} a_n e_n \right| \\ &< (d-1)\frac{T}{2}\alpha + \beta \\ &\leq \alpha \sum_{n \in I_k} a_n + \beta. \end{aligned}$$

By (i) and (ii),

$$\left| \sum_{n \in I_k} a_n e_n \right| < \alpha \sum_{n \in I_k} a_n + \beta$$

for all $k \geq K$. Therefore $\{e_n\}$ is not persistently disturbing with respect to $\{a_n\}$. ∎

Condition 3 in the above theorem is a powerful simplification of the regular Kushner–Clark condition, in which we only need to check the convergence of the KC limit for one positive T. With the theorem, we can state a sufficient and necessary version of the Kushner–Clark condition for the convergence of the scalar Robbins–Monro algorithms.

Theorem 4 *Consider the stochastic algorithm*

$$x_{n+1} = x_n - a_n f(x_n) + a_n r_n + a_n \epsilon_n, \tag{6}$$

where x_n, r_n, ϵ_n are real valued random variables, f is a function from $\mathbb{R}$ into $\mathbb{R}$, and $\{a_n\}$ satisfies (SC). Suppose:

1. *There exists a unique point x^* such that $f(x) > 0$ for $x > x^*$ and $f(x) < 0$ for $x < x^*$;*
2. *$|f(x)| \leq K_f$ for all x;*
3. *For all $\delta > 0$, there exists $h_\delta > 0$ such that $|x - x^*| \geq \delta$ implies $|f(x)| \geq h_\delta$;*
4. *$f(x)$ is continuous at $x = x^*$; and*
5. *$\lim_n r_n = 0$ almost surely.*

Then

$$\lim_{n \to \infty} x_n = x^* \text{ almost surely.}$$

if and only if for every $T > 0$,

$$\lim_{n\to\infty}\left(\sup_{n<p\le m(n,T)}\left|\sum_{i=n+1}^{p} a_i\epsilon_i\right|\right) = 0 \text{ almost surely,}$$

where $m(n,T) \triangleq \max\{k : a_n + \cdots + a_k \le T\}$.

4 Conclusion

We proved the equivalence of the Kushner–Clark condition with the persistent disturbing criterion proposed by Kulkarni and Horn [5]. The necessity of the latter condition implies that the widely applied Kushner–Clark condition is in fact necessary and therefore cannot be relaxed. We should point out that the convergence of KC limit is *in probability* in the original form of the Kushner–Clark theorem, instead of *almost surely* in the presented form. However, the equivalence between this two forms of convergence can be easily established with the necessity of the almost surely convergence condition. For future research, we wish to weaken the conditions placed upon the functions. Furthermore, we hope to explore the connections between the probabilistic characteristics of the noise sequence and the two almost surely conditions discussed here.

References

[1] H. Robbins and S. Monro, "A stochastic approximation method," *Annals of Mathematical Statistics*, vol. 22, pp. 400–407, 1951.

[2] H. K. Kushner and D. S. Clark, *Stochastic Approximation Methods for Constrained and Unconstrained Systems.* New York: Springer, 1978.

[3] M. Metivier and P. Priouret, "Applications of a Kushner and Clark lemma to general classes of stochastic algorithms," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 140–151, Mar. 1984.

[4] D. S. Clark, "Necessary and sufficient conditions for the Robbins-Monro method," *Stochastic Processes and their Applications*, vol. 17, pp. 359–367, 1984.

[5] S. R. Kulkarni and C. Horn, "Necessary and sufficient conditions for convergence of stochastic approximation algorithms under arbitrary disturbances." Submitted, 1993.

AN ALGORITHM FOR TRACKING AND DATA ASSOCIATION BY SINGULARITY ESTIMATION [1]

A. Satish, Rangasami L. Kashyap and S. Malhotra
School of Electrical Engineering
Purdue University
West Lafayette, IN 47907

ABSTRACT : A new method (Algorithm for Data Association (ADA)) using intercept point forecasts is proposed to maintain association between direction of arrival (DOA) estimates of different targets. At each time instant, 't', the intercept point forecast information of the intersecting targets obtained from existing data is employed for data association. Data comprise DOA estimates of targets ($\hat{\theta}$) obtained from the stochastic maximum likelihood estimation method. The forecasted intercept point is recognized as the estimated singularity of a single second order curve of the form $\hat{\theta}^2 = \alpha_0 t^2$ fitted to the data, where the parameter α_0 characterizes the curve. The estimated singularity of the second order polynomial is given by a recursive least squares solution. Tracking of direction of arrival of targets is achieved by successive DOA estimation, intercept point forecasting and data association.

I. INTRODUCTION

A. Objective : There has been considerable interest recently in developing efficient algorithms for tracking direction of arrival (DOA) of multiple moving targets [1-4 and references therein] by utilizing noisy measurements of target signals impinging on a uniform linear array of passive sensors. Important applications include, air-traffic control in the vicinity of crowded airports wherein an efficient tracking algorithm would be invaluable for tracking incoming, outgoing and circling aircraft. In more recent developments, phased array technology is being incorporated at base-stations of a cellular communications network. This enhances capability of locating and tracking of mobiles and also provides the opportunity to efficiently utilize the available radio spectrum, thus increasing overall capacity of the network [5]. In view of these applications, the objective is to develop an efficient algorithm to track direction of arrival of multiple moving targets.

B. Data Association : One of the most *important* issues in multiple target tracking is the measurement/target 'data association' problem : that is, determining which measurement in the measurement set corresponds to which target. In sensor array based methods for angle target tracking, data association implies association of DOA estimates of different targets at two successive time instants and we term it 'estimate association'.

For the case of D targets, state model based methods like Probabilistic Data Association (PDA) [6], track splitting [7], involve searching over the D! possible combinations. In [3], a dynamic model governing the motion of the targets is used. The maximum likelihood (ML) DOA estimates [8] obtained at each interval are refined using a Kalman filter wherein angular velocity and acceleration of the targets (components of the state vector in addition to the DOA) are also estimated. A necessary condition for correct estimate association in [2, 3] requires the targets to have different signal powers. In the recent approach of [4], a new ML criterion function is formulated for joint estimation of DOA and range for a large sensor array. This is based on an approximation on the inverse of the data covariance matrix for large M (# sensors) in the original stochastic ML criterion function [8]. From this ML method, an algorithm for tracking multiple targets called the Tracking ALgorithm (TAL) is developed. The formulation of the likelihood function is such that a structure is imposed for ML estimation of range and angle parameters of each of D targets. With this, the estimates obtained at every instant of time are naturally *ordered.* Thus,

[1]This work was supported by the Innovation Science and Technology (IST) Program of the BMDO monitored by the Office of Naval Research under contract ONR N00014-91-J-4126.

estimate association is automatically achieved by associating the respective 2-tuples (range & angle estimates) of the targets at any two successive time instants.

In this paper, a new method for estimate association is proposed wherein the idea is to utilize the forecast of the point of intersection (*intercept point* (IP)) of *any two* intersecting targets to assign that particular DOA estimate to the correct target. The *key* observation is, the intercept point can be represented by the singularity point of a second order curve fitted to data comprising of angle estimates of the two targets. If D targets are present, D(D-1)/2 intercept point estimates are obtained. Since $D > D(D-1)/2$ for $D > 3$, a method is proposed to suitably combine these intercept point estimates to achieve correct association.

Thus, the overall angle of arrival tracking algorithm consists of three parts -- estimating DOAs from array data, using estimated DOAs as data for forecasting the D(D-1)/2 intercept points and then combining the forecasts to perform estimate association.

II. OVERVIEW OF PROPOSED METHOD

A. Motivation for intercept point forecasting :

To begin with, the targets are assumed not to have crossed each other. For targets moving far apart, associating estimates is straightforward, nearest neighbors at successive time instants are associated (initial part of the track in Fig. 1). However, at cross-over, estimates are closely spaced and additional information is crucial for correct association. The *important* idea is to recognize that intercept point forecast information of the intersecting targets at each tracking time instant, can indeed be used for *estimate association*, i.e., cross-over of target tracks is detected from the nature of intercept point forecast. The DOA estimates over several tracking time instants form *data* for forecasting the intercept point. The intercept point is completely specified by the DOA angle at interception called *Intercept Angle* and time of interception called *Intercept Time* (Fig. 1).

The intercept point can be represented as the *singularity* of a second order curve of the form $\hat{\theta}^2 = \alpha_0 t^2$ which is fitted to available data (see Fig. 1). '$\hat{\theta}$' is the DOA estimate and 't' is the tracking time. α_0 characterizes the second order curve. This forms the *second* step of the developed algorithm. A recursive weighted least squares procedure, the Algorithm for Estimation of Singularity (AES), is formulated for estimating the intercept point co-ordinates.

The *final* step is the Algorithm for Data Association (ADA) procedure for associating the tracks correctly. Depending on the position of the forecasted intercept point, a decision of whether the targets have crossed each other or not is made and DOA estimates are assigned to respective target tracks.

B. Tracking Algorithm Overview :

Step 1 DOA estimation of D targets at every time instant using the stochastic ML method. Details about the signal model and the estimation method are given in [8].

Step 2 A forecast of the intercept point between every pair of targets is obtained by fitting a *single* curve of the form $\hat{\theta}^2 = \alpha_0 t^2$ to 'n' ($n \in Z^+$) angle estimates of each of the two targets available at that time instant and estimating the singularity. The method is termed 'Algorithm for Estimation Singularity' (AES) and is described in section III. In this manner, D(D-1)/2 intercept point forecasts are obtained.

Step 3 Using D(D-1)/2 intercept point forecasts obtained at time $t = t_1$, the D unlabeled DOA estimates at current time instant $t = t_1$ are associated to the labeled estimates at previous time instant $t = t_1 - 1$. The proposed algorithm is called 'Algorithm for Data Association (ADA)' and is given in section IV.

III. ALGORITHM FOR ESTIMATION OF SINGULARITY (AES)

The objective is to fit a single curve to every pair of *estimated* tracks (data) to estimate the intercept point. At every time instant, $t = t_1$, we get D unlabeled DOA estimates. The strategy is to associate each of the D unlabeled samples to their nearest neighbor from the previous estimated tracks to form tentative tracks. These tentative tracks selected two at a time form data for intercept point estimation. The second order curve chosen to fit data from any two tracks indexed by $k = k_1, k_2$, $k_1, k_2 \in \{1, 2, .., D\}$ is

$$\Lambda: F(t,\hat{\theta}) = \hat{\theta}^2 - t^2 = 0 \tag{3.1}$$

where 't' represents tracking time, '$\hat{\theta}$' is the DOA estimate, $\hat{\theta}, t \in \Re$. Though DOA estimates are obtained only at discrete time instants, $\hat{\theta}$ and t are continuous, and take values on the real line. All operations are restricted to I quadrant of the $t - \hat{\theta}$ plane, since for a linear array $\hat{\theta} \in (0, \pi)$ and $t \geq 0$. The general form of the chosen curve suggests a constant acceleration assumption on the target dynamics. The singularity lies on the origin of the $t - \hat{\theta}$ plane in (3.1). However, within the I quadrant, the position of the intercept point is arbitrary. Hence, the general form in (3.1) needs to be translated. Equation (3.1) is also subjected to a rotational transformation to accommodate for arbitrary orientations of the trajectories. The translated and rotated curve parameterizing the data is given by

$$\left[(\hat{\theta} - \theta^*)\cos\beta - (t - T^*)\sin\beta\right]^2 = \alpha_0\left[(t - T^*)\cos\beta + (\hat{\theta} - \theta^*)\sin\beta\right]^2 \tag{3.2}$$

where θ^* = true intercept angle, T^* = true intercept time, α_0 = parameter describing the curve, β = angle of rotation of the curve.

We need to solve for the four parameters $[T^*, \theta^*, \alpha_0, \beta]$ in (3.2). Eqn. (3.2) is rewritten as

$$\hat{\theta}^2 = A\hat{\theta} + Bt^2 + Ct + Dt\hat{\theta} + E \text{ where}$$

$$A = 2\theta^* - \frac{T^*(\sin 2\beta)(1+\alpha_0)}{\cos^2\beta - \alpha_0\sin^2\beta},\quad B = -\frac{\sin^2\beta - \alpha_0\cos^2\beta}{\cos^2\beta - \alpha_0\sin^2\beta},\quad D = \frac{(\sin 2\beta)(1+\alpha_0)}{\cos^2\beta - \alpha_0\sin^2\beta}$$

$$C = \frac{2T^*\left(\sin^2\beta - \alpha_0\cos^2\beta\right) - \theta^*(\sin 2\beta)(1+\alpha_0)}{\cos^2\beta - \alpha_0\sin^2\beta} \text{ and}$$

$$E = \frac{\theta^*T^*(\sin 2\beta)(1+\alpha_0) - \left(T^{*2} - \theta^{*2}\alpha_0\right)\sin^2\beta - \left(\theta^* - T^*\alpha_0\right)\cos^2\beta}{\cos^2\beta - \alpha_0\sin^2\beta} \tag{3.3}$$

From (3.3), five parameters are needed to fit the curve in (3.2). We define the parameter vector to be $\psi_t \equiv \text{col.}[E_t, C_t, B_t, D_t, A_t]$. Hence, for $2t > 5$, the over determined system of 2t equations with 5 unknowns that is to be solved to obtain $\hat{\psi}_t$ is given by $z_t = \Xi_t\psi_t$. (3.4)

At time $t = t_1$, we have the data $\left\{\left\{\hat{\theta}^{k_1}_{(t)}\right\}, \left\{\hat{\theta}^{k_2}_{(t)}\right\}, t_0 \leq t \leq t_1\right\}$ from two tentative tracks formed from angle estimates obtained at discrete tracking time instants in the interval $t_0 \leq t \leq t_1$. We assume initial value $t_0 = 1$ and the tracking time interval to be of unit length. Thus, there are t_1 2-vectors forming the data. The resulting system is given by

$$\underbrace{\begin{bmatrix} \left(\hat{\theta}^{k_1}_{t_0}\right)^2 \\ \left(\hat{\theta}^{k_2}_{t_0}\right)^2 \\ . \\ \left(\hat{\theta}^{k_1}_{t_1}\right)^2 \\ \left(\hat{\theta}^{k_2}_{t_1}\right)^2 \end{bmatrix}}_{z_{t_1}(2t_1 \times 1)} = \underbrace{\begin{bmatrix} 1 & t_0 & t_0^2 & t_0\hat{\theta}^{k_1}_{t_0} & \hat{\theta}^{k_1}_{t_0} \\ 1 & t_0 & t_0^2 & t_0\hat{\theta}^{k_2}_{t_0} & \hat{\theta}^{k_2}_{t_0} \\ 1 & . & . & . & . \\ 1 & t_1 & t_1^2 & t_1\hat{\theta}^{k_1}_{t_1} & \hat{\theta}^{k_1}_{t_1} \\ 1 & t_1 & t_1^2 & t_1\hat{\theta}^{k_2}_{t_1} & \hat{\theta}^{k_2}_{t_1} \end{bmatrix}}_{\Xi_{t_1}(2t_1 \times 5)} \underbrace{\begin{bmatrix} E_{t_1} \\ C_{t_1} \\ B_{t_1} \\ D_{t_1} \\ A_{t_1} \end{bmatrix}}_{\psi_{t_1}(5\times 1)} \tag{3.5}$$

Solving the linear system in (3.5), we get $\hat{\psi}_{t_1}$ which is the estimated parameter vector ψ_t at time t_1. Knowing $\hat{\psi}_{t_1}$, we solve the system in (3.3) to get $\{\hat{T}_{t_1}, \hat{\phi}_{t_1}\}$, the intercept point estimates of the true intercept point $\{T^*, \theta^*\}$ at time t_1. Thus, from (3.3), we get

$$C / B = -2T^* - \theta^*(D / B) \tag{3.6}$$

$$C / D = -2T^*(B / D) - \theta^* \tag{3.7}$$

Solving the simultaneous equations in (3.6) & (3.7), (*linear* in θ^*, T^*),

$$\hat{T}_{t_1} = -\frac{2\hat{C}_{t_1} + \hat{A}_{t_1}\hat{D}_{t_1}}{4\hat{B}_{t_1} + \hat{D}^2_{t_1}} \tag{3.8}$$

and $$\hat{\phi}_{t_1} = \left(\frac{4\hat{B}_{t_1} + \hat{D}^2_{t_1}}{2\hat{A}_{t_1}\hat{B}_{t_1} - \hat{C}_{t_1}\hat{D}_{t_1}} \right) \tag{3.9}$$

Remark 3.1 : We *do not* solve for $\alpha_{0_{t_1}}$ and $\beta_{0_{t_1}}$ since we are only interested in intercept point estimation. Estimation of $\alpha_{0_{t_1}}$ and $\beta_{0_{t_1}}$ involves solving a non-linear system of equations.

The Recursive Least Squares (RLS) approach [9] is used to estimate the $\hat{\psi}_t$ vector and the corresponding estimate at time t_1 is denoted by $\hat{\psi}_{t_1}$. We have a vector observation at every time instant, i.e., we have a pair of DOA estimates, one from each target. The RLS recursions are given by :

$$\hat{\psi}_t = \hat{\psi}_{(t-1)} + L_t[z_t - \xi_t^T \hat{\psi}_{(t-1)}] \tag{3.10}$$

$$L_t = Q_t^{-1}\xi_t = P_{(t-1)}\xi_t\left[\lambda_t I + \xi_t^T P_{(t-1)}\xi_t\right]^{-1} \tag{3.11}$$

$$P_t = \frac{1}{\lambda_t}\left[P_{(t-1)} - L_t\xi_t^T P_{(t-1)}\right] \tag{3.12}$$

$$P_t = Q_t^{-1} \tag{3.13}$$

z_t is the vector observation of length 2. L_t is the 5 x 2 gain matrix, the two columns representing the two targets. λ_t is called the 'forgetting factor' which discounts the effect of the older measurements exponentially. We choose $\lambda_t = \lambda$. λ is always assumed to be less than one to ensure that the gain will not reduce to zero. Hence, $Q_t = \sum_{k=1}^{t} \lambda^{t-k}\xi_k\xi_k^T$ where ξ_t is a 5 x 2 regression matrix. Eqns (3.8-3.13) summarize the AES algorithm.

IV. ALGORITHM FOR DATA ASSOCIATION (ADA)

At time t_1, we get an *un-associated* D-vector of DOAs (also called measurement vector). The task is to associate this vector to an already *associated* D-vector of DOAs (at time (t_1-1)) as shown in Fig. 2. The following four issues are of concern :

(a) Forming initial estimated tracks for starting recursion (Algorithm AA).
(b) Selecting appropriate data for intercept point estimation (Algorithm BB).
(c) Cross-over detection using intercept point forecasts (Algorithm CC).
(d) Evidence combination of D(D-1)/2 intercept point forecasts to establish the D links between successive time instants (Algorithm DD).

A. Forming initial estimated tracks (Algorithm AA):

To start with, the targets are assumed to be away from each other. At every time instant upto a certain time t_{LS}, the new un-associated D-vector of angle estimates is linked to the previous associated D-vector based on their order statistics, i.e., the highest is associated with the highest, the second highest with the second highest and so on, thus forming the initial portion of the tracks. This is defined as *straight linking*.

B. Selection of appropriate measurement vectors (Algorithm BB) :

For recursive intercept point estimation at each time instant $t = t_1$, $t_1 > t_{LS}$, 2n angle estimates (n for each of two targets chosen) are used for fitting the curve and estimating the intercept point, out of which 2(n-1) have been previously associated and two which still have to be associated (Fig. 2(a)). The number (n) of DOA estimates (data points) to get a good fit is greater than (5x2 = 10) because at least five points are needed to estimate ψ_t (3.5). The

factor '2' accounts for selection of two targets at a time. For D targets, two targets can be chosen in D(D-1)/2 different ways. Thus, we get D(D-1)/2 intercept point estimates.

The *key* hypothesis is, for intercept point estimation at time t_1, the unlabeled DOA estimates at time t_1 are *not* associated, but just *straight linked* to the associated ones at time (t_1-1). Two cases are to be justified :

Targets far away : *If* the targets are *away* from each other, this is the most natural thing to ensure correct association to get accurate forecasts of the intercept point.

Targets in proximity : *If* the targets are *close*, i.e., the DOA estimates are very close to each other, *straight linking* will *still* ensure that the intercept point estimates are accurate, because, the DOAs are so close that it *does not* matter which one we are going to pick for determining the recursive intercept point forecast.

After selecting the appropriate 'n' estimates of every pair of estimated tracks, D(D-1)/2 intercept point forecasts are obtained using AES (section III).

C. Cross-over detection from intercept point forecast (Algorithm CC):

1) Idea behind Algorithm CC :

Consider angle estimates from estimated tracks of any two targets moving towards each other as in Fig. 2(a). A new vector of unlabeled measurements are obtained at time t_1. There exist two possible associations as seen from Figs. 2(b) and 2(c). We need to choose one of two hypotheses to determine the appropriate linking using the intercept point forecast. For the two targets under consideration, the following three situations can occur (Fig. 3).

(i) If two targets are approaching (Fig. 3(a)), the intercept point forecast is *in front* of the latest measurement vector (at time t_1).

(ii) If the two targets are at crossover (Fig. 3(b)), the intercept point forecast is *in between* the 2 measurement vectors (at times (t_1-1) and t_1).

(iii) If the two targets are diverging (moving away) from each other (Fig. 3(c)), the intercept point forecast is *behind* both measurements.

This observation forms the basis for cross-over detection.

2) Algorithm CC:

Check for the intercept point forecast in the region enclosed by measurements at times (t_1-1) and t_1. If intercept point is present, cross-over is detected, hence cross-link DOAs (Fig. 4(b)). If intercept point is not present, cross-over is not detected, hence straight link DOAs (Fig. 4(a) & 4(c)).

Remark 4.1 : The intercept point forecast from the RLS fit should accurately depict each situation. Experimental results confirm accuracy of RLS estimates of point of intersection. More importantly, at time t_1, we have D estimates and D(D-1)/2 intercept points, i.e., the number of intercept points is greater than number of targets. Thus, information of all intercept points should be combined to decide upon the D links between (t_1-1) and t_1.

D. Evidence combination for linking (Algorithm DD):

In view of remark 4.1, the current objective becomes one of linking D estimates of targets at successive time instants using D(D-1)/2 intercept point forecasts. An algorithm is proposed to handle situations wherein *more than one* intercept point occurs within one tracking interval or multiple intercept points exist in *successive* tracking instants.

DD1) Structure of the link matrix :

Every element of the D-vector of DOA estimates at time (t_1-1) is associated and labeled because the measurement vectors at times (t_1-1) and (t_1-2) are associated. This labeling information is stored in a *link* matrix, $LM(t_1-1)$ of dimension Dx2.

The first column of the link matrix consists of DOA estimates arranged in descending order of magnitude and the second column identifies the measurements, i.e., it contains the target label. The second column of the link matrix is called *index vector*. One possible link matrix configuration at time (t_1-1) is given by

$$LM(t_1-1)=\begin{bmatrix}\hat{\theta}^{1}_{(t_1-1)} & \hat{\theta}^{2}_{(t_1-1)} & \cdot & \hat{\theta}^{k}_{(t_1-1)} & \cdot & \hat{\theta}^{D}_{(t_1-1)}\\ 2 & 1 & . & k & . & D\end{bmatrix}^{T} \text{ where } \hat{\theta}^{1}_{(t_1-1)} \geq \hat{\theta}^{2}_{(t_1-1)} \geq \hat{\theta}^{k}_{(t_1-1)} \geq .. \geq \hat{\theta}^{D}_{(t_1-1)} \quad (4.1)$$

Without loss of generality, the link matrix at the first time instant, LM(t = 1) consists of the DOA estimates arranged in descending order with the first estimate getting the label of target '1' and the last one getting label 'D'.

DD2) Formation of the Ordering Matrix :

Based on intercept point forecasts at time t_1 and the link matrix, $LM(t_1-1)$ at time $(t_1\text{-}1)$, a DxD symmetric matrix called *Ordering Matrix*, $OM(t_1)$ is formed. The elements of this matrix contain integer valued target labels. The upper and lower triangular elements give ordering information for association of measurements at t_1 with the already associated ones at $(t_1\text{-}1)$. The ith row and column contain information about the ith target interacting with the other (D-1) targets. Labels to D^2 entries of $OM(t_1)$ are assigned as follows :

- $OM_{(t_1)}(i,i) = i,\ i = 1, 2, ..., D.$ (4.2)
- For assigning labels to off-diagonal elements, the intercept point forecasts and the index vector of the link matrix (4.1) are used. The order in which these D(D-1)/2 elements are filled is determined by choosing indices (i,j) of elements $OM_{(t_1)}(i,j)$, $(i \neq j)$ as follows :

  Two elements i and j of the index vector are chosen at a time, beginning from the first row and proceeding downwards till the (D-1)th row. The entries of $OM(t_1)$ are

  $OM_{(t_1)}(i,j) = OM_{(t_1)}(j,i) = i$, if cross - over is detected (4.3a)

  $OM_{(t_1)}(i,j) = OM_{(t_1)}(j,i) = j$, if no cross - over is detected (4.3b)

  where $(i,j) \in \{1,2,\ldots,D\}$.

Remark 4.2 : $\hat{\theta}^{i}_{(t_1-1)} \geq \hat{\theta}^{j}_{(t_1-1)}$, $(i,j) \in \{1,2,\ldots,D\}$ since the DOA column vector in the link matrix is always sorted and arranged in descending order.

Remark 4.3: The detection of cross-over which is used as a basis for ordering is assumed to be accurate which is indeed a crucial assumption (see Remark 4.1).

The D required links are established from the link matrix as illustrated below.

DD3) Forming the link matrix from the ordering matrix :

- The first column of the link matrix, $LM(t_1)$ consists of unlabeled DOA estimates of time t_1 arranged in descending order.
- The ith row of the reordering matrix represents the ith target. The number of non-zero entries obtained by taking absolute difference between elements of the ith row and a row vector consisting 'i's (index i) gives the order statistic of the ith target. This procedure is repeated for all rows and the *labels* of targets are obtained. Thus, the index vector of the link matrix at time t_1 is formed. The above insertion sort procedure completes estimate association between angle estimates between times $(t_1\text{-}1)$ and t_1.

V. PROPOSED TRACKING ALGORITHM

For every time instant, $t = 1,2,\ldots,(t_1-1), t_1, (t_1+1),\ldots,\hat{T}$ where $\hat{T}$ = last instant of tracking period, repeat the following steps (1) to (4).

(1) From a sequence of N snapshots, $\{y(\tau)\}_{\tau=1}^{\tau=N}$, estimate direction of arrival parameters $\{\hat{\theta}^{k}_{(t)}\}_{k=1}^{k=D}$, $\hat{\Gamma}_{ML}$, $\hat{\rho}_{ML}$ of the D targets by the stochastic ML method [8].

(2) For every pair of targets [D(D-1)/2 pairs],

(a) Using respective DOA estimates, obtain $\hat{\psi}_t$ from recursions in AES (Eqns (3.10)-(3.13)).

(b) Substitute elements of $\hat{\psi}_t$ in (3.8) & (3.9) to compute forecasted intercept point $\{\hat{T}_t, \hat{\phi}_t\}$.

(3) Associate the estimates using forecasted intercept points by ADA procedure (section IV).

(4) Use associated estimates as initial guesses for DOA estimation at succeeding time instant.

Initialization :

For initializing the recursion in AES, wait until time say, $t_k \geq 5$ when P_{t_k} becomes invertible. $\hat{\psi}_{t_k}$ is then computed using the least squares estimate

$$\hat{\psi}_{t_k} = [\Xi^*_{t_k} \Xi_{t_k}]^{-1} \Xi^*_{t_k} z_{t_k} \tag{5.1}$$

The entire data $\left\{\hat{\theta}_t^{k_1}, \hat{\theta}_t^{k_2}\right\}_{t=t_0}^{t=t_k}$ upto time t_k where $k_1, k_2 \in \{1,2,\ldots,D\}$ & $t_0 \equiv 1$ is used.

VI. PERFORMANCE OF PROPOSED METHOD

A. Experimental Results :

Experiment 1 : We consider the simplest two target case with only one intercept point. Targets move rapidly with rate of change of DOA being 3^0 /s. The chosen SNR is -3 dB with number of sensors = 8. Forty snapshots are used to estimate the 2 DOAs at every sampling instant. The proposed algorithm is able to track the two targets correctly (Fig. 5(a)). Figures 5(b) and (c) show forecasts of intercept angle and time at every tracking instant respectively. It is observed that, the intercept time estimate (21.2s) converges close to the true value (21.1613s), so that the tracks flip exactly one time instant after true cross-over. The key point is, the values converge to the neighborhood of their true values rapidly enough facilitating correct detection of target intercept and subsequent tracking.

The RLS intercept point estimates are fairly accurate even at low SNR and with small number of sensors. Accuracy of the intercept point forecasts depends on the accuracy of the DOA estimates (data). Since the ML method gives very accurate angle estimates, the data used for curve fitting is essentially outlier free. In effect, the effective SNR of data used for obtaining intercept point forecasts is very high, though the array data itself can be very noisy. The chosen SNR and number of sensors are thresholds for the two target case.

Experiment 2 : There are 4 targets describing linear trajectories. The number of sensors is 16 and SNR is still at 0 dB. Although three targets cross each other at *almost* the same time, the algorithm is still able to track the trajectories correctly (Fig. 6). The intercept point estimates are accurate and come in correctly at cross-over and hence the data association algorithm ensures that appropriate angle estimates are linked. Even if the DOA estimates are not accurate at cross-over, the accuracy of the intercept point estimates is not affected. This is one of the important features of the proposed method.

B. Discussion : A computationally simple and efficient solution to the data association problem is proposed. The number of sensors (M) in the uniform linear array is the *key* factor for good performance of the algorithm. Increasing the number of sensors causes intercept point forecasts to converge rapidly to their true values. This is because, increasing M increases effective SNR and hence highly accurate DOA estimates are obtained which in turn are data for estimating the intercept point. If M is sufficient, the algorithm works well even at low SNR (0 , -3 dB). The requirement of the number of sensors increases with increase in number of targets. Typically, the algorithm requires 16-20 sensors for upto six targets.

SNR is not a significant factor for accurate estimate association. The number of snapshots required for DOA estimation at every time instant is around 40 (even with 4-6 targets) indicating good performance of the angle estimation method. The fitted curve accounts for rotation and translation transformations and hence the method works well for arbitrarily oriented tracks. The proposed algorithm can be implemented partly in parallel, the intercept point forecasting for the D(D-1)/2 track pairs can be done simultaneously.

The second order fit of the angle track data implies the assumption of constant acceleration on target motion which is reasonable. The method does not require apriori knowledge about targets (e.g. SNR) identity to perform intercept point prediction. Even if we entirely miss either target bearing due to jamming signals, the prediction of the intercept point will not deteriorate rapidly since we are fitting a single curve to both angle tracks, i.e., the prediction is robust to errors in tracking. Also, instead of fitting a *single* curve, if we fit *two* regression lines, the line can completely flip in presence of a few outlier points. It is likely for this problem to occur if DOA estimates are incorrectly assigned.

By intercept point forecasting, we reduce the overall computational complexity of estimate association to D(D-1)/2 comparisons instead of D^2. For e.g., for D = 10, D^2 = 100 whereas D(D-1)/2 = 45. Since we estimate D(D-1)/2 intercept points every time (assuming constant number of targets), the worst case complexity of proposed method is also D(D-1)/2.

VII. CONCLUSION

A real-time recursive algorithm was developed to effectively track the angle of arrival of multiple moving targets based on intercept point estimation. The intercept point estimates are accurate and converge rapidly to the true values. However, performance of the algorithm is significantly dependent on the accuracy of intercept point forecast. In situations where the targets are very fast and not linear, the RLS forecast takes a finite amount of time to converge and if targets cross *before* convergence, the cross-over is not detected. The proposed algorithm could also be used for tracking slowly varying sinusoids which cross each other in additive noise, like the first formant frequency in speech signals, bio-medical signals etc.

REFERENCES

[1] C. K. Sword, M. Simaan, and E. W. Kamen, "Multiple Target Angle Tracking using Sensor Array Outputs," *IEEE Trans. AES*, **26**, No. 2, 1990, 367-373.

[2] C. R. Sastry, E.W. Kamen, and M. Simaan, "An efficient Algorithm for Tracking Angles of Arrival of Moving Targets, " *IEEE Trans. SP*, **39**, No. 1, 1991, 242-246.

[3] C. R. Rao, C. R. Sastry, and B. Zhou, "Tracking the Direction of Arrival of Multiple Moving Targets, " *IEEE Trans. SP*, **42**, No. 5, 1994, 1133-1145.

[4] A. Satish and R. L. Kashyap, "Multiple Target Tracking using Maximum Likelihood Principle," *In review, IEEE Trans. SP*, Nov. 1993.

[5] Soren Anderson, Mille Millnert, et al, "An Adaptive Array for Mobile Communication Systems," *IEEE Trans. on Veh. Tech.*, **40**, No. 1, 1991, 230-236.

[6] Y. Bar-Shalom & T. E. Fortmann, *Tracking and Data Association*, AP, Fl., 1988.

[7] P. Smith and G. Buechler, "A Branching Algorithm for Discriminating and Detecting Multiple Objects, " *IEEE Trans. Auto. Control.* **20**, No. 2, 1975, 101-104.

[8] J. F. Bohme, "Estimation of Spectral Parameters of Correlated Signals in Wavefields," *Signal Processing*, **10**, 1986, 329-337.

[9] L. Ljung, *System Identification - Theory for the user*, Prentice - Hall, NJ, 1987.

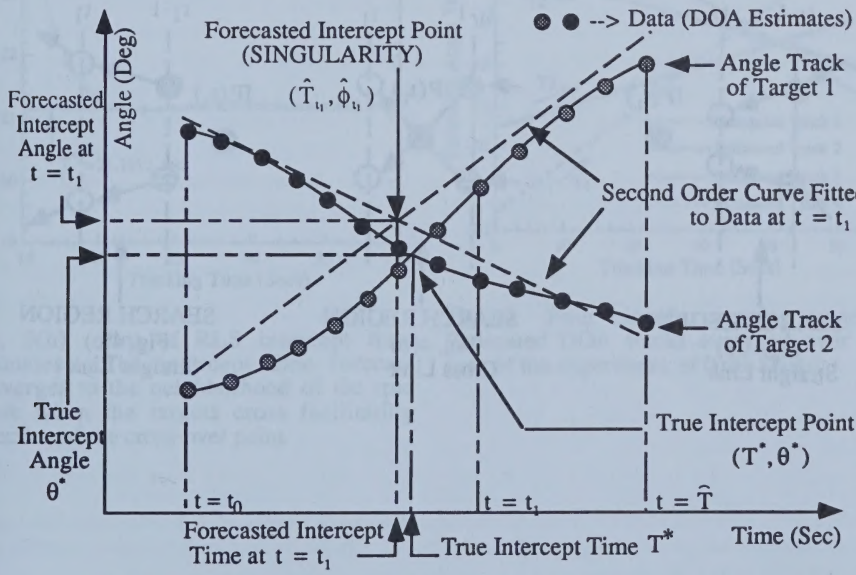

Forecasted Intercept Point
(SINGULARITY)
$(\hat{T}_{t_1}, \hat{\phi}_{t_1})$
--> Data (DOA Estimates)
Angle Track of Target 1
Angle (Deg)
Forecasted Intercept Angle at $t = t_1$
Second Order Curve Fitted to Data at $t = t_1$
Angle Track of Target 2
True Intercept Angle θ^*
True Intercept Point (T^*, θ^*)
$t = t_0$
$t = t_1$
$t = \hat{T}$
Forecasted Intercept Time at $t = t_1$
True Intercept Time T^*
Time (Sec)

Fig. 1 Illustration of the idea behind Intercept Point Estimation .

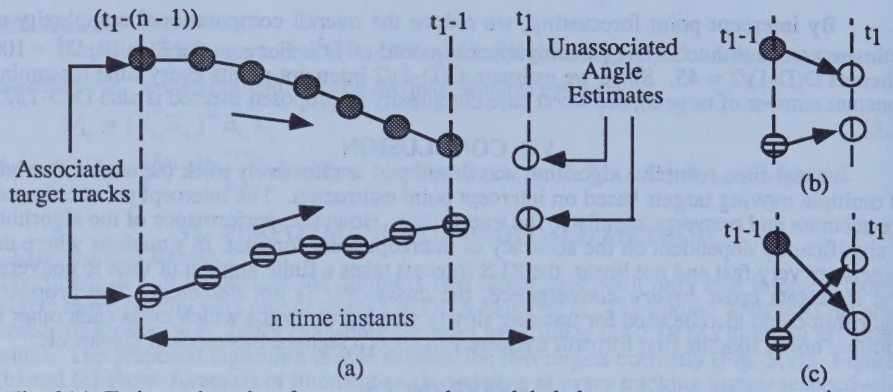


Fig. 2(a) Estimated tracks of two targets and an unlabeled measurement vector at time t_1.
Fig. 2(b) & 2(c) The two possible associations between estimates at times (t_1-1) and t_1.

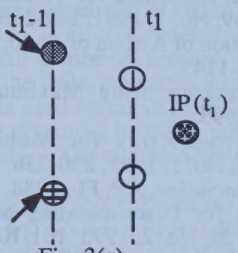


Fig. 3(a)
Targets Approaching
each other

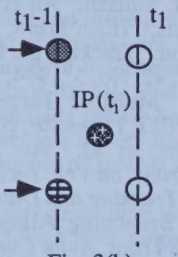


Fig. 3(b)
Targets at cross-over

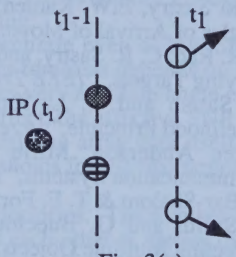


Fig. 3(c)
Targets diverging
after cross-over

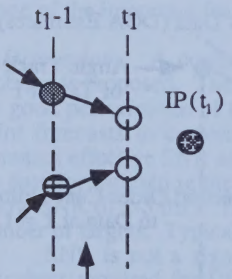


Fig. 4(a)
Straight Link

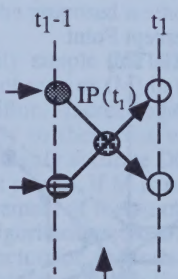


Fig. 4(b)
Cross Link

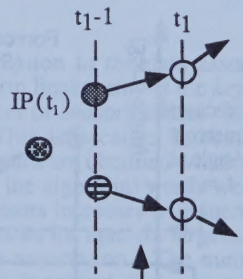


Fig. 4(c)
Straight Link

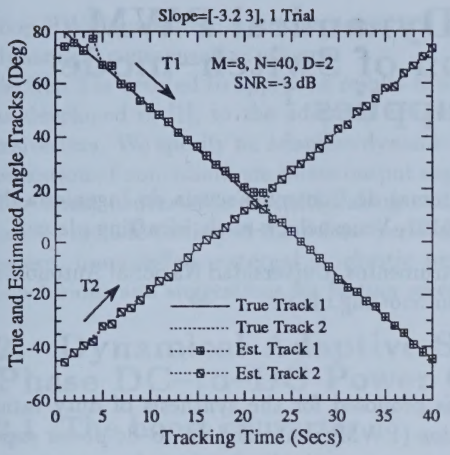


Fig. 5(a) Plot of true & estimated trajectories.

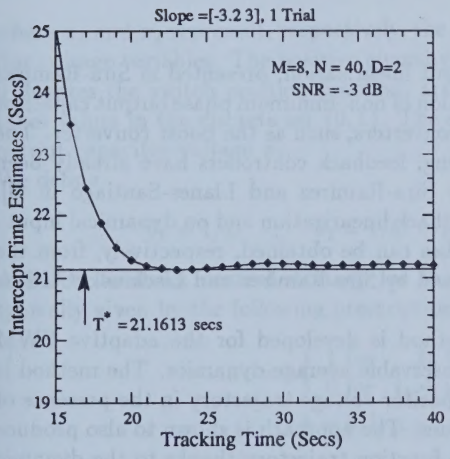


Fig. 5(b) Plot of RLS intercept time estimates. The intercept time forecast converges to the neighborhood of the true value when the targets cross facilitating detection of the cross-over point.

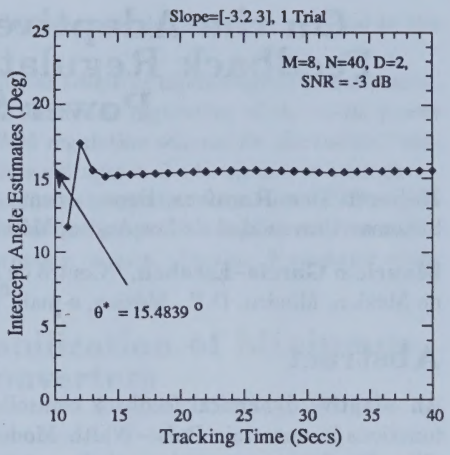


Fig. 5(c) Plot of RLS intercept angle estimates. The intercept angle forecasts are accurate and also converge to the neighborhood of the true value facilitating detection of cross-over point.

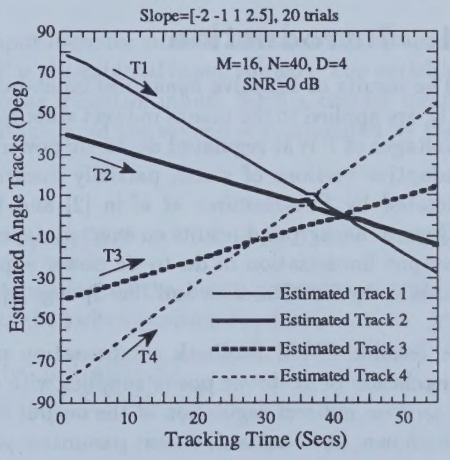


Fig. 6 Four closely crossing targets. Estimated DOA tracks averaged over 20 runs of the experiment at 0 dB SNR.

On the Adaptive Dynamical PWM Feedback Regulation of Switch–mode Power Supplies [1]

Hebertt Sira–Ramírez, Departamento Sistemas de Control, Escuela de Ingeniería de Sistemas Universidad de Los Andes, Mérida 5101, Venezuela, e-mail: isira@ing.ula.ve,

Mauricio García–Esteban, [2] Centro de Instrumentos, Universidad Nacional Autónoma de México, México, D.F., México, e-mail: mauricio@ing.ula.ve

Abstract

An adaptive dynamical feedback controller is proposed for the synthesis of duty ratio functions in uncertain Pulse–Width–Modulation (PWM) controlled dc–to–dc power supplies. The input–output dynamic linearizing, overparametrized, adaptive scheme is shown to provide indirect output capacitor voltage stabilization by means of suitable direct regulation of the minimum–phase input inductor current. A design example is included dealing with the boost converter circuit in continuous conduction mode. The robustness of the approach is tested via computer simulations which incorporate large, unmatched, external stochastic input signals of bounded amplitude.

1. Introduction

The results of adaptive *dynamical* input–output linearization, presented in Sira-Ramírez [1], are applied to the case of indirect stabilization of non–minimum phase output capacitor voltages of PWM regulated dc–to–dc power converters, such as the boost converter. The adaptive versions of *static*, partially linearizing, feedback controllers have already been treated by Sira-Ramírez *et al* in [2] and by Sira-Ramírez and Llanes-Santiago in [3]. General background results on exact state feedback linearization and on dynamical input–output linearization of dc–to–dc power supplies can be obtained, respectively, from the articles by Sira-Ramírez and Ilic–Spong, [4], and by Sira-Ramírez and Lischinsky-Arenas [5].

A feasible PWM feedback compensation method is developed for the adaptive PWM regulation of dc–to–dc power supplies with observable average dynamics. The method is based on indirect regulation of the output capacitor voltage trajectory in the presence of unknown, but constant, circuit parameter values. The approach is shown to also produce a substantially smoothed feedback duty ratio function trajectory thanks to the dynamic nature of the adaptive linearizing duty ratio generator.

The proposed approach suitably combines three important features : 1) Linearizing compensation, based on full, and not partial, exact input–output linearization, 2) The overall scheme is adaptive and, hence, assumes no knowledge of the converter parameters, 3) The proposed adaptive duty ratio synthesizer (controller) is dynamical in nature. Hence, the oscillatory nature of the duty ratio function trajectory, arising from the actual closed

[1]This research was supported by the Consejo de Desarrollo Científico, Humanístico and Tecnológico of the Universidad de Los Andes under Research Grant I–456–94, by the Consejo Nacional de Ciencia y Tecnología (CONACYT), México

[2]On temporary leave from the Instituto de Cibernética, Universidad Politécnica de Catalunya, Barcelona, España

loop PWM operation, is further smoothed out by the natural integration involved in the dynamical compensation scheme.
Section 2 is devoted to apply the results of nonlinear adaptive input–output linearization, as developed in [1], to the adaptive dynamical feeedback regulation of dc–to–dc power converters. We specify an adaptive dynamic PWM regulation scheme for the indirect stabilization of non–minimum phase output capacitor voltages in dc–to–dc power converters. The boost converter is presented along with extensive computer simulations. The simulations include testing of the robust performance of the proposed adaptive scheme under severe, unmodelled, external stochastic perturbation inputs. Section 3 contains some conclusions and suggestions for further research.

2. Dynamical Adaptive Stabilization of Minimum Phase DC–to–DC Power Converters

2.1 The boost converter

Consider the boost converter model shown in figure 1. This circuit is described by the following state equation model,

$$\begin{aligned} \dot{x}_1 &= -(1-u)\, x_2/L + E/L \\ \dot{x}_2 &= (1-u)\, x_1/C - x_2/RC \\ y &= x_1 \end{aligned} \tag{2.1}$$

where x_1 and x_2 represent, respectively, the input inductor current and the output capacitor voltage variables. The positive quantity E is the external input voltage. The variable u denotes the switch position function, acting as a control input. Such a control input takes values in the discrete set $\{0, 1\}$. The output, y, of the system is represented by the output capacitor voltage x_2.
We define

$$\theta_1 = 1/L \quad ; \quad \theta_2 = 1/C \quad ; \quad \theta_3 = 1/RC \quad ; \quad \theta_4 = E/L \tag{2.2}$$

A PWM feedback control strategy for the regulation of the boost converter circuit is typically given by the following prescription of the switch position function,

$$\begin{aligned} u &= \begin{cases} 1 \text{ for } & t_k \le t < t_k + \mu[x(t_k)]T \\ 0 \text{ for } & t_k + \mu[x(t_k)]T \le t < t_k + T \end{cases} \\ & k = 0, 1, \ldots \end{aligned} \tag{2.3}$$

where t_k represents a sampling instant, the parameter T is the fixed sampling period, also called the *duty cycle*, the sampled values of the state vector $x(t)$ of the converter are denoted by $x(t_k)$. The function, $\mu(\cdot)$, is the *duty ratio function* acting as a truly feedback policy. The value of the duty ratio function, $\mu[x(t_k)]$, determines, at every sampling instant, t_k, the width of the upcomming "pulse" as $\mu[x(t_k)]T$. The actual duty ratio function, $\mu(\cdot)$, is, evidently, a function limited to the closed interval, $[0, 1]$, of the real line. The control problem, associated to the stabilization of the discontinuously controlled system, (2.1),(2.3), towards some (feasible) prespecified constant equilibrium point, consists in specifiying the duty ratio function, μ, *as a feedback policy*, i.e., as a function of the state vector x. As formulated, the problem of synthesizing a suitable duty ratio function $\mu(x)$ is quite involved, due to the difficulty in performing an exact discretization of the nonlinear PWM controlled circuit model (2.1),(2.3) (see Kassakian *et al* [6]).

A conceptually useful and rather practical alternative consists in resorting to the *Infinite Frequency Average PWM model*, of the switch regulated system. Such an average model is also known as *State Space Average Model* (see Middlebrook and Ćuk [7]). The assumption of an infinite sampling frequency results in a smooth nonlinear average system model of (2.1) in which the duty ratio function, μ, is readily interpreted as a control input to the average system. Such an interpretation results in the formal replacement of the switch position function u by the duty ratio function μ in the averaging process. In fact, the duty ratio function becomes the *equivalent control* input in the corresponding *Sliding Mode* (see Utkin [8]) interpretation of the obtained idealization (see also Sira-Ramírez [9]). An extensive use of the described infinite frequency average models, with successful results, has been made in a series of articles dealing with nonlinear feedback regulation policies for dc–to–dc power supplies (see references [10]–[12])
The average PWM model of system (2.1),(2.3) is then described by,

$$\begin{aligned} \dot{\zeta}_1 &= -\theta_1(1-\mu)\,\zeta_2 + \theta_4 \\ \dot{\zeta}_2 &= \theta_2(1-\mu)\,\zeta_1 - \theta_3\zeta_2 \\ y &= \zeta_1 \end{aligned} \tag{2.4}$$

where the averaged state variables are denoted by ζ_1 and ζ_2. Note that we still, abusively, denote the average output voltage by the same variable y.
The constant equilibrium point of the average system dynamics (2.4), corresponding to a constant value U, $(0 < U < 1)$, of the duty ratio function μ, is given by

$$\mu = U \;\; ; \;\; \zeta_1 = Z_1(U) = \frac{\theta_3\theta_4}{\theta_2\theta_1(1-U)^2} \;\; ; \;\; \zeta_2 = Z_2(U) = \frac{\theta_4}{\theta_1(1-U)} \tag{2.5}$$

The equilibrium values $Z_1(U)$ and $Z_2(U)$ are, both, positive quantitities.
The observability matrix (see Conte *et al* [13]) associated to the output, $y = \zeta_1$, of the average system (2.4) is given by

$$\mathcal{O} = \begin{bmatrix} 1 & 0 \\ 0 & -\theta_1(1-\mu) \end{bmatrix} \tag{2.6}$$

The rank of the matrix is everywhere equals to 2, except when μ is identically 1. This condition physically corresponds to a permanently *saturated* duty ratio function, thus entitling a fixed switch position ($u \equiv 1$) and, hence, no possibilities exist of feedback regulation.
Consider now the input–output description of the average PWM controlled boost converter (see [5]),

$$\ddot{y} = -\theta_1\theta_2(1-\mu)^2 y - \frac{\dot{y}-\theta_4}{1-\mu}\left(\theta_3(1-\mu) + \dot{\mu}\right) \tag{2.7}$$

The *zero dynamics*, associated to the constant equilibrium point of the system, $y = Z_1(U)$, $\dot{y} = 0$, is given by

$$\dot{\mu} = \frac{\theta_3}{(1-U)^2}(1-\mu)(\mu-U)\;(\mu+U-2) \tag{2.8}$$

The equilibrium points of the zero dynamics are clearly given by $\mu = U$, $\mu = 1$ and $\mu = 2 - U$. As the phase diagram of system (2.8) reveals (see Figure 2), the equilibrium point $\mu = 1$ is unstable while the other two are asymptotically stable. The only physically meaningful equilibrium point is given by $\mu = U$, since $\mu = 2 - U$ corresponds to a duty ratio larger than 1.

Suppose it is desired to regulate the input inductor current x_1 towards a desired constant value $X_1(U) = Y$, modulo small chattering due to the pulsed regulatory actions of the actual PWM feedback controller. It has been rigorously shown in [9] that the actual PWM regulated state trajectories can be made to converge, in an arbitrarily close fashion, towards the average closed loop PWM controlled state responses, provided a sufficiently high value of the sampling frequency $1/T$, associated to every PWM regulation scheme, is used. Our PWM stabilization problem is, therefore, primarily solved on the basis of the average PWM model (2.4), as if such a model accurately represented the actual PWM controlled system (2.1),(2.3).
Let the desired asymptotically stable closed loop linear dynamics for the average PWM controlled converter be of the form:

$$\ddot{y} + 2\xi\omega_n\dot{y} + \omega_n^2(y - Y) = 0 \tag{2.9}$$

The corresponding dynamical feedback regulator which synthesizes the required *computed* duty ratio function μ, is written, in transformed variables ζ_1 and ζ_2, as

$$\dot{\mu} = \frac{1}{\theta_1\zeta_2}\left[-\omega_n^2(\zeta_1 - Y) + 2\xi\omega_n\theta_1(1-\mu)\zeta_2 - 2\xi\omega_n\theta_4 + \theta_1\theta_2(1-\mu)^2\zeta_1 - \theta_1\theta_3(1-\mu)\zeta_2\right] \tag{2.10}$$

If the circuit parameters $\theta_1, \ldots, \theta_4$ are unknown then the adaptive version of the dynamic linearizing controller (2.10) is specified as,

$$\dot{\mu} = \frac{1}{\hat{\theta}_1\zeta_2}\left[-\omega_n^2(\zeta_1 - Y) + 2\xi\omega_n\hat{\theta}_1(1-\mu)\zeta_2 - 2\xi\omega_n\hat{\theta}_4 + \widehat{\theta_1\theta_2}(1-\mu)^2\zeta_1 - \widehat{\theta_1\theta_3}(1-\mu)\zeta_2\right] \tag{2.11}$$

In terms of the *overparametrization* vector $\hat{\Theta}$ (see Campion and Bastin [14]), defined as,

$$\Theta = [\Theta_1, \ldots, \Theta_{14}] = [\theta_1, \ldots, \theta_4, \theta_1^2, \theta_1\theta_2, \ldots, \theta_2^2, \ldots, \ldots, \theta_3\theta_4, \theta_4^2]^T \tag{2.12}$$

the dynamical feedback controller (2.11) is rewritten as,

$$\dot{\mu} = \frac{1}{\hat{\Theta}_1\zeta_2}\left[-\omega_n^2(\zeta_1 - Y) + 2\xi\omega_n\hat{\Theta}_1(1-\mu)\zeta_2 - 2\xi\omega_n\hat{\Theta}_4 + \hat{\Theta}_6(1-\mu)^2\zeta_1 - \hat{\Theta}_7(1-\mu)\zeta_2\right] \tag{2.13}$$

Note that the controller (2.13) exhibits a singularity at $\zeta_2 = 0$, which may be considered to be far away from the desired equilibrium point (2.5). Initial conditions for the system, and the controller state, can always be appropriately set such that this singularity is conveniently avoided.
Following the developments in [1] (see also Sastry and Isidori [15]), we obtain the following expression for the closed–loop behavior of the output variable y,

$$\ddot{y} + 2\xi\omega_n\dot{y} + \omega_n^2(y - Y) = [\tilde{\Theta}_1, \ldots, \tilde{\Theta}_{14}]\begin{bmatrix} W_1(\zeta, \hat{\Theta}, \mu) \\ \vdots \\ W_{14}(\zeta, \hat{\Theta}, \mu) \end{bmatrix} \tag{2.14}$$

The equations constituting the parameter estimation error update law are summarized as follows:

$$\dot{\tilde{\Theta}}_i = -\dot{\hat{\Theta}}_i = -\gamma_i\frac{e_1\vartheta_i}{1 + \vartheta^T\vartheta} \quad ; \quad i = 1, \ldots, 14 \tag{2.15}$$

where,

$$\begin{aligned} W_i(\zeta,\hat{\Theta},\mu) &= 0 \;\; ; \;\; i = 2,3,5,8,\ldots,14 \\ W_1(\zeta,\hat{\Theta},\mu) &= -2\xi\omega_n(1-\mu)\zeta_2 + \zeta_2\dot{\mu} \\ W_4(\zeta,\hat{\Theta},\mu) &= 2\xi\omega_n \\ W_6(\zeta,\hat{\Theta},\mu) &= -\zeta_1(1-\mu)^2 \\ W_7(\zeta,\hat{\Theta},\mu) &= -\zeta_2(1-\mu) \end{aligned} \tag{2.16}$$

and,

$$e_1 = \sum_{i=1}^{14} \tilde{\Theta}_i \vartheta_i = y - Y - \sum_{i=1}^{14} \hat{\Theta}_i \vartheta_i + \frac{1}{s^2 + 2\xi\omega_n s + \omega_n^2} * \left[\sum_{i=1}^{14} \hat{\Theta}_i W_i(\zeta,\hat{\Theta},\mu) \right] \tag{2.17}$$

The filtered regressor vector components ϑ_i are obtained as solutions of the following differential equations, with zero initial conditions,

$$\begin{aligned} \ddot{\vartheta}_i &= -2\xi\omega_n\dot{\vartheta}_i - \omega_n^2\vartheta_i + W_i(\zeta,\hat{\Theta},\mu) \\ & \quad i = 1,4,6,7 \end{aligned} \tag{2.18}$$

Evidently,

$$\vartheta_i = 0 \quad \text{for} \;\; i = 2,3,5,8,\ldots,14 \tag{2.19}$$

The adaptive dynamically synthesized duty ratio function, (2.13),(2.15)–(2.19), is now easily translated into an acutal PWM regulation scheme of the form (2.3), by means of high frequency sampling of the on–line genarated (computed) duty ratio function trajectory $\mu(t)$. Figure 3 depicts the proposed adaptive dynamical PWM feedback regulation scheme. Note that in such a feedback scheme, rather than using the average values of the state vector ζ_1, ζ_2, as demanded by the expression (2.13), the actual PWM regulated values of the state vector components, x_1, x_2, are used, for feedback purposes. The above PWM feedback control technique has been fully justified in [9]– [12] for a variety of nonlinear feedback regulation strategies.

2.2 Simulation Results

Simulations were carried out for a boost converter regulated by the proposed nonlinear adaptive dynamical feedback controller specified by (2.13),(2.15)–(2.19). The following parameter values were used in the simulation model, but otherwise they were assumed to be unknown: $L = 0.020$ H, $C = 20$ μF $R = 30$ Ω and $E = 15$V. The equilibrium value for the duty ration function, generated by the dynamical controller, was set to be $\mu = U = 0.6$. The corresponding "nominal" equilibrium point, for the average input inductor current and average output capacitor voltage, were taken to be , respectively, $\zeta_1 = Y = Z_1 = 3.125$Amp, $\zeta_2 = Z_2 = 37.5$Volts. The dynamically generated duty ratio function μ was used on a PWM controller of the form (2.3), with a sampling frequency of 5 KHz.

In order to test the robustness of the proposed adaptive feedback control scheme, the designed dynamical controller was used on a "perturbed" version of the boost converter. For this purpose a zero mean, computer generated, bounded stochastic signal η, was allowed to act as an unmodelled external perturbation input η to the converter in the following manner,

$$\begin{aligned} \dot{x}_1 &= -\theta_1(1-u)x_2 + \theta_4 + \eta \\ \dot{x}_2 &= \theta_2(1-u)\,x_1 - \theta_3\,x_2 \\ y &= x_1 \end{aligned} \tag{2.20}$$

The peak to peak value of the "unmatched" perturbation signal, η, was set to be approximately 15% of the assigned simulation value used for the parameter E/L.
The desired second order asymptotically stable linear dynamics, (2.9), for the regulated input inductor current $y = x_1$ was characterized by the following parameters $\xi = 0.80$ $\omega_n = 500$.
The simulations, shown in Figure 4, depict the behavior of the nonlinear dynamical PWM regulated state trajectories $x_1(t)$, $x_2(t)$ for the parameter uncertain, externally perturbed, converter model (2.20). The smooth evolution of the dynamically generated duty ratio function, $\mu(t)$, is shown along with a small portion of the corresponding PWM regulated switch position trajectory $u(t)$.
Only the trajectories of the components of the vector of estimated parameters, $\hat{\Theta}(t)$, exhibiting significant variations from their arbitrarily assigned initial values, are shown in this figure. The adaptation gains γ_i ; $i = 4, 5, 6, 7$ were set to be $\gamma_1 = \gamma_4 = 9.0 \times 10^6$ and $\gamma_6 = \gamma_7 = 1$. Finally, a small portion of the, computer generated, external stochastic perturbation input signal η is depicted in the figure.
The PWM controlled state variables, x_1 and x_2, are seen to asymptotically converge towards the required equilibrium values, modulo the expected chattering associated to PWM regulation. In spite of the "unmatched" nature of the applied perturbation input, and its significant amplitude values, the system state variables are shown to exhibit remarkable robustness with respect to such an external perturbation signal. The control objective of indirectly driving the converter output capacitor voltage to a prespecifed equilibrium value with desirable stability features for the regulated variable is seen to be accomplished in a rather satisfactory manner.

3. Conclusions and Suggestions for Further Research

In this article a stable adaptive feedback control scheme has been proposed for the dynamical stabilization of minimum phase output signals in parameter uncertain dc–to–dc power supplies. The scheme constitutes an extension of the results given in [15], and by Sasty and Bodson in [16]. The developments, however, include the case in which the underlying linearizing feedback controller is dynamical.
The results were applied to the indirect regulation of non–minimum phase output capacitor voltages in a PWM controlled dc–to–dc power supply, such as the boost converter for which only static adaptive regulation schemes were available.
The proposed adaptive feedback regulation scheme is primarily based on the idealized infinte frequency sampling model, or state average model, of the PWM regulated converter. In spite of the idealization, the actual discontinuously controlled trajectories are seen to follow remarkably close the average closed loop trajectories for modestly high sampling frequencies. The scheme is also quite robust with respect to unmodelled external stochastic perturbation signals of bounded nature.
The problem of direct regulation of non–minimum phase output signals continues to be a challenging problem, specially for the case of parameter uncertain systems. In the switched mode dc–to–dc power converters case, the output capacitor voltage signal constitutes a non–minimum phase output on which the proposed regulation scheme is not applicable. In a sense, our approach of indirect output voltage regulation through stabilization of the input inductor current, already proposed in [5], also constitutes an application of the method recently proposed by Benvenuti *et al* in [17] (see also Fliess and Sira–Ramírez, [18] for a module theoretic justification of the method in linear systems).

References

[1] H. Sira-Ramírez, M. Zribi and S. Ahmad, "Adaptive Dynamical Feedback Regulation Strategies for Linearizable Uncertain Systems," *International Journal of Control.* Vol. 57, No. 1, pp. 121-139, 1993.

[2] H. Sira-Ramírez, R. Tarantino-Alvarado and O. Llanes-Santiago, " Adaptive Feedback Stabilization in PWM Controlled DC-to-DC Power Supplies," *International Journal of Control* Vol. 57, No. 3, pp. 599-625, 1993.

[3] H. Sira-Ramírez y O. Llanes-Santiago, " Adaptive PWM Regulation Schemes in Swiched Controlled Systems" *Proc. of the 12th World Congress of the IFAC*, Sydney, Australia, Vol. 10, pp. 57–60, 1993.

[4] H. Sira-Ramírez and M. Ilic, " Exact Linearization in Switch Mode DC-to-DC Power Converters, " *International Journal of Control.* Vol. 50, No. 2, pp. 511-524, 1989.

[5] H. Sira–Ramírez and P. Lischinsky–Arenas, "Differential algebraic approach in non–linear dynamical compensator design for dc–to–dc power converters", *International Journal of Control*, Vol. **54**, No. 1, pp. 111–133, 1991.

[6] J. G., Kassakian, M. F., Schlecht, and G. C., Verghese, *Principles of Power Electronics*, Addison Wesley Publishing Co., Reading, Massachusetts, 1991.

[7] R.D. Middlebrook and S. Ćuk, "A General Unified Approach to Modeling Switching Converter Power Stages" in *International Journal of Electronics* Vol. 42, No. 6, pp. 521–550, 1977.

[8] V.I., Utkin, *Sliding Modes and Their Applications in Variable Structure Systems*, MIR Publishers, Moscow 1978.

[9] H. Sira-Ramírez, " A Geometric Approach to Pulse-Width-Modulated Control in Nonlinear Dynamical Systems," *IEEE Transactions on Automatic Control.* Vol. AC-34, No.2, pp. 184-187, 1989.

[10] H. Sira-Ramírez, "Switched Control of Bilinear Converters via Pseudolinearization, " *IEEE Transactions on Circuits and Systems.* Vol. CAS-36, No. 6. pp. 858-865, 1989.

[11] H. Sira-Ramírez y M.T. Prada-Rizzo, "Nonlinear Feedback Regulator Design for the Cuk Converter," *IEEE Transactions on Automatic Control* Vol. AC-37, No. 8, pp. 1173-1180, 1992.

[12] H. Sira-Ramírez, "Nonlinear P-I Controller Design for Switch-mode DC-to-DC Power Converters," *IEEE Transactions on Circuits and Systems.* Vol CAS-38, No. 4, pp. 410-417, 1991.

[13] G. Conte, C. Moog and A. Perdon,"Un théoreme sur la repersentation entrée–sortie d'un système non linéaire", *Comptes Rendus de l'Academie des Sciences de Paris*, Serie I, vol. 307, pp. 363–366, 1988.

[14] G. Campion and G. Bastin, "Indirect Adaptive State Feedback Control of Linearly Parametrized Non–linear Systems," *International Journal of Adaptive Control and Signal Processing*, Vol 4., No. 3, pp. 345–358, 1990.

[15] S. Sastry and A. Isidori, "Adaptive Control of Linearizable Systems", *IEEE Transactions on Automatic Control*, Vol. AC–34, No. 11, pp. 1123–1131, 1989.

[16] S. Sastry and M. Bodson, *Adaptive Control: Stability, Convergence and Robustness*, Prentice Hall, Englewood Cliffs, New Jersey, 1989.

[17] L. Benvenuti, M. D. Di Benedetto, J. W. Grizzle, "Approximate output tracking for nonlinear non–minimum phase systems with applications to flight control," Report CGR–92–20, Michigan Control Group Reports. University of Michigan, Ann Arbor Michigan, U.S.A. 1992.

[18] M. Fliess and H. Sira-Ramírez, "Regimes glissants, structures variables linéaires et modules, *C. R. de la Acad. de Sci. Paris.* Serie I, Automatique. pp. 703–706, 1993

FIGURES

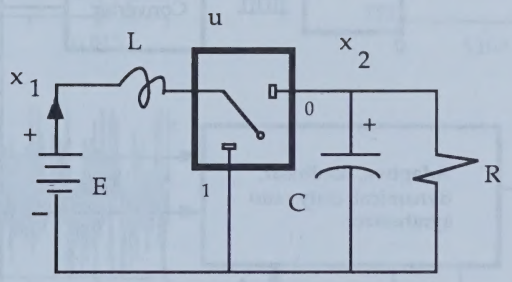


Figure 1. The boost converter.

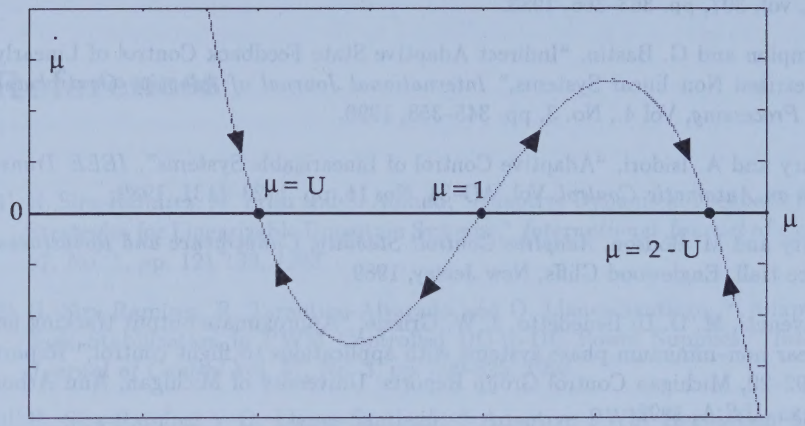


Figure 2. Zero dynamics for the boost converter.

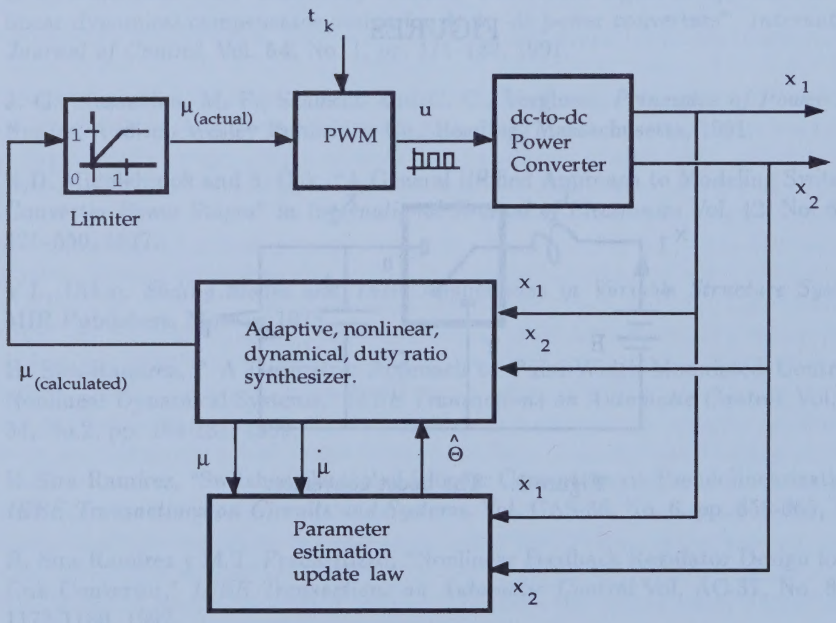


Figure 3. An adaptive dynamical PWM feedback control

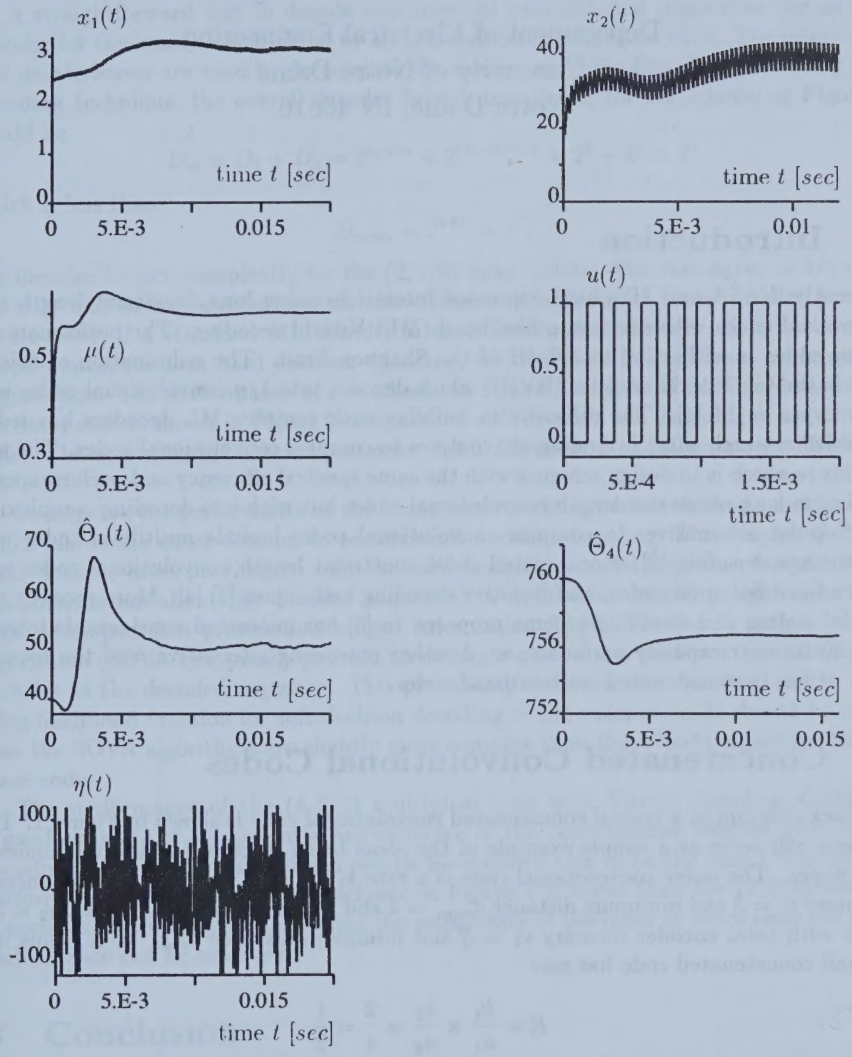


Fig. 4. Actual nonlinear adaptive PWM controlled state trajectories, duty ratio function trajectory, time evolution of the switch positions, trajectories of the estimated parameters and external perturbation signal for the boost converter example.

Alternatives to Complex Convolutional Codes*

Lance C. Perez
Daniel J. Costello, Jr.

Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46616

1 Introduction

Recently, NASA and JPL have expressed interest in using long constraint length convolutional codes with maximum likelihood (ML) Viterbi decoding. The performance of these codes is within 2.0 to 3.0 dB of the Shannon limit. The culmination of this effort is the big Viterbi decoder (BVD), which decodes rate $1/n$ convolutional codes with memory $\nu = 14$ [1]. The difficulty in building such complex ML decoders has led to renewed research effort in finding alternatives to complex convolutional codes. The goal of this research is to design schemes with the same spectral efficiency and performance as optimum long constraint length convolutional codes but with less decoding complexity.

Popular alternatives to complex convolutional codes include multilevel codes with multistage decoding [2], concatenated short constraint length convolutional codes with outer Reed-Solomon codes, and iterative decoding techniques [3]-[4]. More recently, the Turbo coding and decoding scheme proposed in [5] has generated considerable interest due to its near capacity performance. Another promising alternative, and the focus of this paper, is concatenated convolutional codes.

2 Concatenated Convolutional Codes

A block diagram of a typical concatenated convolutional code is shown in Figure 1. This system will serve as a simple example of the ideas being discussed in the remainder of the paper. The outer convolutional code is a rate $k_1/n_1 = 2/3$ code with total encoder memory $\nu_1 = 3$ and minimum distance $d_{min_1} = 4$ and the inner code is a rate $k_2/n_2 = 3/4$ code with total encoder memory $\nu_1 = 3$ and minimum distance $d_{min_2} = 4$. Thus, the overall concatenated code has rate

$$R = \frac{k_1}{n_1} \times \frac{k_2}{n_2} = \frac{2}{4} = \frac{1}{2}$$

and the sum of the total encoder memories is $\nu_1 + \nu_2 = 6$. This example was chosen as a possible alternative to to the CCSDS standard $(2, 1, 6)$ convolutional code with minimum distance $d_{min} = 6$ [6].

An important fact concerning concatenated binary convolutional codes is that the overall minimum distance of the code is not the product of the individual minimum

*This work was supported by NASA grants NAG 5-557 and NAG 3-1549.

distances. If the generator matrices of the two codes of our example are multiplied, the result is a $(4,2,7)$ code with $d_{min} = 6$ and $d_{next} = 9$. This is much less than $d_{min_1} \times d_{min_2} = 16$. The performance of the multiplied $(4,2,7)$ code with soft decision Viterbi decoding is shown in Figure 2. As expected, the multiplied code performs ≈ 0.6dB worse than the $(2,1,6)$ code due its smaller minimum distance.

A straightforward way to decode concatenated convolutional codes is to use an ML decoder for the inner code followed by an ML decoder for the outer code. The interleaver and deinterleaver are used to decorrelate the errors out of the first decoder. Using this decoding technique, the overall decoder branch complexity for the scheme of Figure 1 would be

$$D_{cc} = D_1 + D_2 = 2^{k_1+\nu_1} + 2^{(k_2-1)+\nu_2} = 2^5 + 2^5 = 2^6$$

which is less than

$$D_{ccsds} = 2^{k+\nu} = 2^7,$$

the decoder branch complexity for the $(2,1,6)$ code. (Note: The rate $k_2/n_2 = 3/4$ code has only four branches entering each state, since one information bit is uncoded.)

However, if a standard ML decoder is used for the first decoder, then the second decoder must perform hard decision decoding, which results in significant performance degradation. The performance of the scheme in Figure 1 with hard decision decoding of the outer code is shown in Figure 2 compared to soft decision ML decoding of the $(2,1,6)$ code. The concatenated scheme performs ≈ 3.0 dB worse at a bit error rate (BER) of 10^{-5}.

To improve the performance of the concatenated scheme, some method must be used which allows the outer decoder to perform soft decision decoding. This implies that the inner decoder must provide not only the hard decision decoded sequence, but also some soft information about the decoded sequence. A number of decoding algorithms have been developed that provide soft output information [7]-[9]. These soft output Viterbi algorithms (SOVA's) typically provide an estimate p_k of the *a posteriori* probability for each bit in the decoded sequence. This probability estimate may then be used to form a log-likelihood function for soft decision decoding of the outer code. It should be noted that the SOVA algorithms are slightly more complex than the Viterbi algorithm for the same code.

The performance of the $(4,2,7)$ multiplied code with Viterbi decoding should be a good estimate of the potential performance of the concatenated scheme with SOVA decoding. (Preliminary simulation results have shown this to be the case.) Though the performance of the concatenated scheme of Figure 1 is worse than the $(2,1,6)$ code, it is possible that by slightly increasing the complexity of the inner or outer code the same performance can be achieved.

3 Conclusion

A number of new coding techniques may result in practical schemes with reduced decoding complexity and with performance approaching capacity. The key element in these techniques is the ability to pass soft decision information between concatenated, pipelined, or parallel maximum likelihood decoders. Of particular interest are concatenated convolutional codes with soft output decoding of the inner code. Work is currently underway on finding a concatenated convolutional coding scheme that outperforms the optimal $(2,1,14)$ code.

References

[1] O. M. Collins, "The subleties and intricacies of building a constraint length 15 convolutional decoder," *IEEE Trans. Commun.*, **COM-40**, pp. 1810–1819, 1992.

[2] S. Lin, "Low complexity and high performance concatenated coding schemes for high speed satellite and space communications," *Technical Report 93-004* to NASA Goddard Space Flight Center, August 1993.

[3] E. Paaske, "Improved decoding for a concatenated coding system recommended by CCSDS," *IEEE Trans. Commun.*, **COM-38**, pp. 1138–1144, 1990.

[4] O. M. Collins and M. Hizlan, "Determinate state convolutional codes," *IEEE Trans. Commun.*, **COM-41**, pp. 1785–1794, 1993.

[5] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," Proc. 1993 IEEE Int. Conf. on Comm., Geneva, Switzerland, pp. 1064–1070, 1993.

[6] Consultative Committee for Space Data Systems, *Recommendation for Space Data System Standards: Telemetry Channel Coding*, May 1984,

[7] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, **IT-20**, pp. 284–287, 1974.

[8] L. N. Lee, "Real-time minimal-bit-error probability decoding of convolutional codes," *IEEE Trans. Commun.*, **COM-30**, pp. 146-151, 1974.

[9] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," in *Proc. 1993 IEEE GLOBECOM '89*, Dallas, Texas, pp. 47.1.1–47.1.7, 1989.

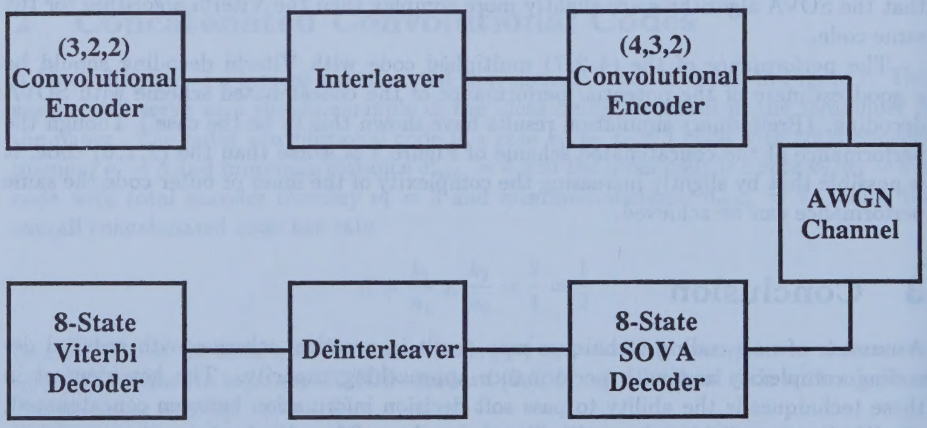


Figure 1: Typical Concatenated Convolutional Coding Scheme

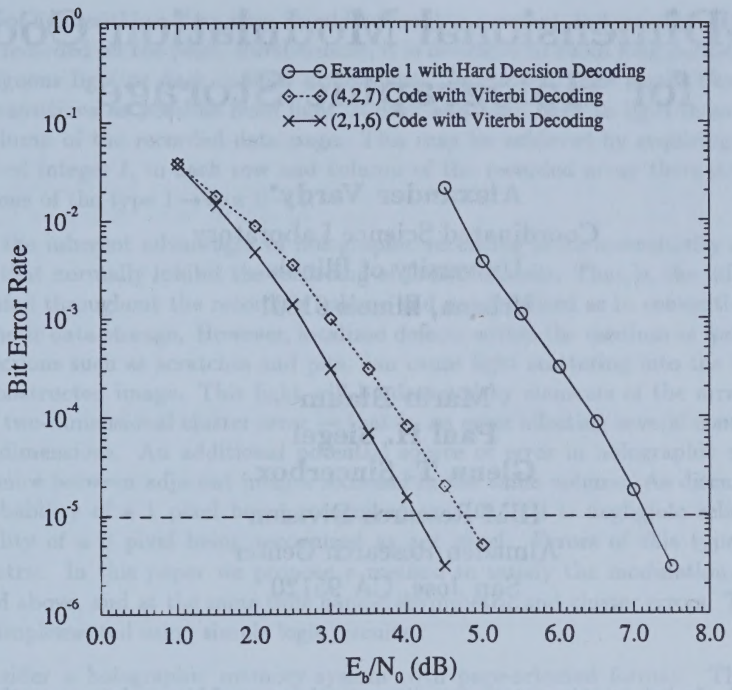


Figure 2: Performance of Example 1

Two-Dimensional Modulation Codes for Holographic Storage

Alexander Vardy*
Coordinated Science Laboratory
University of Illinois
Urbana, Illinois 61801

Mario Blaum
Paul H. Siegel
Glenn T. Sincerbox
IBM Research Division
Almaden Research Center
San Jose, CA 95120

Abstract. In holographic storage, two-dimensional arrays of binary data are optically recorded in a medium via an interference process. To ensure optimum operation of a holographic recording system, it is desirable that the patterns of 1's (light) and 0's (no light) in the recorded array satisfy the following modulation constraint: in each row and column of the array there are at least t transitions of the type $1 \rightarrow 0$ or $0 \rightarrow 1$, for a prescribed integer t. It is also desirable to have the total number of 1's in the recorded array equal to the total number of 0's. We present an algorithm for encoding unconstrained binary data into two-dimensional arrays satisfying these properties. The algorithm employs differential coding, the pigeonhole principle, and error-correcting codes. Using two binary codes – one per dimension – with minimum Hamming distance $d \geq 2t - 3$, we apply a certain transformation to an arbitrary information array which ensures that the number of transitions in each dimension is determined by the minimum distance of the corresponding code.

1. Introduction

In holographic storage two-dimensional data arrays (pages) are optically recorded via an interference process and subsequently retrieved by illumination of the hologram and forming an image of a page on a matched two-dimensional array of photodetectors. To ensure optimum operation of the holographic recording system, the patterns of 1's (light) and 0's (no light) have to satisfy certain modulation constraints. For instance, the total

*This work was supported in part by the National Science Foundation under Grant NCR-9409688.

amount of light within a data page should be roughly constant, independent of the content of data recorded on the page. Furthermore, it is desirable to avoid long periodic stretches of contiguous light or dark in both dimensions. To this end, one would like to have as many transitions as possible from light to dark and from dark to light in each row and each column of the recorded data page. This may be achieved by requiring that, for a prescribed integer t, in each row and column of the recorded array there are at least t transitions of the type $1 \rightarrow 0$ or $0 \rightarrow 1$.

One of the inherent advantages of holographic recording is its insensitivity to localized defects that normally inhibit the recording of individual bits. That is, the information is distributed throughout the recording volume and not localized as in conventional optical or magnetic data storage. However, localized defects within the medium as well as surface imperfections such as scratches and pits, can cause light scattering into the direction of the reconstructed image. This light will be detected by elements of the array and may cause a two-dimensional cluster error — that is, an error affecting several contiguous bits in two dimensions. An additional potential source of error in holographic recording is interference between adjacent images recorded in the same volume. As discussed below, the probability of a 1 pixel being recognized as a 0 pixel is negligible relative to the probability of a 0 pixel being recognized as a 1 pixel. Errors of this type are called asymmetric. In this paper we propose a method to satisfy the modulation constraints specified above, and at the same time handle asymmetric and cluster errors. The method can be implemented using simple logic circuits.

We consider a holographic memory system with page-oriented format. The data are stored in digital form, through a page composer or spatial light modulator, wherein the presence or absence of light at a given bit location represents either a 1 or a 0. Through the process of holographic recording and retrieval, the original light distribution can be subsequently reconstructed and caused to illuminate an array of photodetectors. It has been determined [7] that the principal sources of error in the system are the photon counting errors in the detectors as well as the cross talk errors which are encountered when the storage medium nears its capacity limits. Additional errors may result from light scattered onto the photodetectors by surface imperfections and/or bulk defects.

In order to avoid variations in the ratio between the signal (data) beam and the reference beam during recording and so that there are no wide variations in the intensity of a reconstructed data bit, the number of 1's and 0's in a data page should be balanced. In this way, during recording, the amount of signal light is independent of the data content of a page and, during retrieval, the light coming from a stored page is divided between the same number of beams regardless of the data content.

By requiring this condition to be met exactly, that is requiring that the number of 1's is exactly equal to the number of 0's, additional information about the presence of errors can be realized. Specifically, the balanced property can be combined with two others: the errors tend to occur in two-dimensional clusters because of preferential scattering; it is much more likely that a 0 will be detected as a 1 than conversely. The latter condition occurs because scattered light and/or crosstalk light from adjacent pages can always be detected if it illuminates a bit location that was originally a 0, hence a 0 could become a 1 if there is sufficient noise light. Conversion of a 1 to a 0 requires that the data light and

the noise light interfere destructively; that is, they must have the same amplitude and be 180° out of phase. This is much less likely, thereby giving rise to the error asymmetry. The asymmetric nature of the errors will be detected by the balanced code, which can be used to declare erasures at the affected blocks. The information in the erased blocks can then be retrieved using an interleaved outer code as described in [1].

Although the foregoing discussion pertains to both the modulation and the error-control requirements of holographic storage, in this paper we will focus on coding for the modulation constraints. A more detailed treatment of error-control coding for holographic storage will be presented elsewhere [1, 2].

2. Description of the modulation algorithm

Consider $m \times n$ binary arrays, and assume that the holographic memory page is tiled by such arrays. Then the modulation constraints discussed in the introduction may be stated more precisely in terms of the properties that each such $m \times n$ has to satisfy. For instance, the following property:

P_1 : *The number of* 1*'s in the array is equal to the number of* 0*'s in the array*

ensures that the total amount of light is roughly constant within any sufficiently large rectangular area. Furthermore, tiling the memory page with binary arrays satisfying P_1 enables efficient detection of asymmetric errors. To avoid long periodic stretches of contiguous light or dark we would like to have frequent transitions from light to no light ($1 \rightarrow 0$) and from no light to light ($0 \rightarrow 1$) in both dimensions. Thus, we require the following property:

$P_2(t)$: *In each row and column there are at least* t *transitions of the type* $1 \rightarrow 0$ *or* $0 \rightarrow 1$

where t is an integer greater than 2. In the following we describe an encoding method which maps an arbitrary $m \times n$ binary array $V = [v_{ij}]$, which may be regarded as a sequence of mn information bits, into an array satisfying properties P_1 and $P_2(t)$.

We may employ a horizontal row by row readout to represent V as a vector $\underline{v}$, where

$$\underline{v} = (v_{11}, v_{12}, \ldots, v_{1n},\ v_{21}, v_{22}, \ldots, v_{2n},\ \ldots\ , v_{n1}, v_{n2}, \ldots, v_{mn}) \tag{1}$$

Now consider the set $\mathcal{U}_1 = \{\underline{u}_0, \underline{u}_1, \underline{u}_2, \ldots, \underline{u}_{mn-1}\}$ of control vectors given by:

$$\underline{u}_i = \overbrace{111\cdots111}^{i}\overbrace{000\cdots000}^{mn-i}$$

for $i = 0, 1, \ldots, mn-1$. It is shown in [5, 6] that for any $\underline{v}$ there exists an i such that $\underline{v} \oplus \underline{u}_i$ contains an equal number of 1's and 0's. Thus the $m \times n$ array corresponding to $\underline{v} \oplus \underline{u}_i$,

where $\oplus$ denotes bit by bit summation modulo 2, satisfies property P_1. Notably, the encoding process does not require a conversion table. All the encoder has to do is to complement the bits of V in the horizontal readout order of (1), terminating after the first i bits. The index i can now be encoded in a binary form and appended to the resulting array. Since the cardinality of $\mathcal{U}_1$ is mn, this requires approximately $\lceil \log_2(mn) \rceil$ redundant bits. However, the redundant part should also contain an equal number of 1's and 0's. Therefore, the encoder would require slightly more than $\lceil \log_2(mn) \rceil$ redundant bits.

Example. Let $n = m = 3$. Then the set $\mathcal{U}_1$ consists of 9 elements. These elements, written as 3×3 arrays are given by:

$$\begin{array}{ccccccccc} 0\,0\,0 & 1\,0\,0 & 1\,1\,0 & 1\,1\,1 & 1\,1\,1 & 1\,1\,1 & 1\,1\,1 & 1\,1\,1 & 1\,1\,1 \\ 0\,0\,0 & 0\,0\,0 & 0\,0\,0 & 0\,0\,0 & 1\,0\,0 & 1\,1\,0 & 1\,1\,1 & 1\,1\,1 & 1\,1\,1 \\ 0\,0\,0 & 0\,0\,0 & 0\,0\,0 & 0\,0\,0 & 0\,0\,0 & 0\,0\,0 & 0\,0\,0 & 1\,0\,0 & 1\,1\,0 \end{array} \qquad \square$$

We now construct the set of control arrays $\mathcal{U}_2(t)$, such that for any $m \times n$ array V there exists at least one element $U \in \mathcal{U}_2(t)$, with $V \oplus U$ satisfying property $P_2(t)$. In order to simplify the derivation we shall count the transitions $0 \to 1$ and $1 \to 0$ in a cyclic manner. Thus if a vector begins with a 1 and ends with 0, this would count as a $0 \to 1$ transition. For example there are two $0 \leftrightarrow 1$ transitions in the vector 111000111 and four $0 \leftrightarrow 1$ transitions in the vector 111000110. With this definition of $0 \leftrightarrow 1$ transitions, it is clear that in any vector the number of $0 \to 1$ transitions is equal to the number of $1 \to 0$ transitions. Hence the total number of such transitions must be even.

Let F_n denote the set of all binary n-tuples, and let E_n denote the subset of F_n consisting of all the n-tuples of even weight. For any $\underline{x} = (x_1, x_2, \ldots, x_n) \in F_n$ let $\sigma(\underline{x}) = (x_n, x_1, \ldots, x_{n-1})$ be the cyclic shift of $\underline{x}$. Define a mapping $\phi : F_n \to E_n$ as follows:

$$\phi(\underline{x}) = \underline{x} \oplus \sigma(\underline{x})$$

Note that the mapping ϕ is linear. Furthermore, the number of $0 \leftrightarrow 1$ transitions in $\underline{x}$ is equal to the Hamming weight of $\phi(\underline{x})$ (the action of ϕ on the elements of F_n is analogous to the action of the $1+D$ channel on semi-infinite sequences, or to the action of the differential on continuous functions). Although ϕ is evidently not invertible we can formally define the inverse mapping. Let $\underline{y} = (y_1, y_2, \ldots, y_n) \in E_n$ and $\underline{x} = (x_1, x_2, \ldots, x_n) \in F_n$. Then $\phi^{-1}(\underline{y}) = \underline{x}$ iff $\phi(\underline{x}) = \underline{y}$ and $x_1 = 0$. Since $\phi(\underline{x}) = \phi(\underline{x}')$ implies that $\underline{x}$ and $\underline{x}'$ are complements of each other, it is clear that each element of E_n will have exactly one inverse.

Now let C_1 be a $(t-2)$-error-correcting code of length m, and assume that C_1 contains at least $(n+1)$ codewords of even weight. Similarly, let C_2 be a $(t-2)$-error-correcting code of length n, containing at least $(m+1)$ codewords of even weight. When n and m are powers of 2, such error-correcting codes exist for all $t \leq \min\{n/4, m/4\}$. These are just the first-order Reed-Muller codes of length m and n, respectively (cf. [8]). Now define the sets $\phi^{-1}(C_1)$ and $\phi^{-1}(C_2)$ as follows:

$$\phi^{-1}(C_1) = \{\, \phi^{-1}(\underline{c}) \;:\; \underline{c} \in C_1 \text{ and } |\underline{c}| \equiv 0 \bmod 2 \,\}$$
$$\phi^{-1}(C_2) = \{\, \phi^{-1}(\underline{c}) \;:\; \underline{c} \in C_2 \text{ and } |\underline{c}| \equiv 0 \bmod 2 \,\}$$

where $|\cdot|$ denotes the Hamming weight. Let $\underline{a}_0, \underline{a}_1, \ldots, \underline{a}_n$ be some $(n+1)$ fixed elements of $\phi^{-1}(C_1)$, and let $\underline{b}_0, \underline{b}_1, \ldots, \underline{b}_m$ be some $(m+1)$ fixed elements of $\phi^{-1}(C_2)$.

Proposition 1. *For any $\underline{x} \in F_m$, if $\underline{x} + \underline{a}_i$ has less than t transitions, then $\underline{x} + \underline{a}_j$ has at least t transitions for all $j \neq i$. Similarly, for any $\underline{x} \in F_n$, if $\underline{x} + \underline{b}_i$ has less than t transitions, then $\underline{x} + \underline{b}_j$ has at least t transitions for all $j \neq i$.*

Proof. Otherwise we would have

$$\begin{aligned} d(\phi(\underline{x}), \phi(\underline{a}_i)) &= |\phi(\underline{x}) + \phi(\underline{a}_i)| = |\phi(\underline{x} + \underline{a}_i)| \leq t - 2 \\ d(\phi(\underline{x}), \phi(\underline{a}_j)) &= |\phi(\underline{x}) + \phi(\underline{a}_j)| = |\phi(\underline{x} + \underline{a}_j)| \leq t - 2 \end{aligned} \tag{2}$$

where $d(\cdot, \cdot)$ denotes the Hamming distance, and the last inequality follows from the fact that the Hamming weight of $\phi(\underline{x})$ is equal to the number of $0 \leftrightarrow 1$ transitions in $\underline{x}$. Since $\phi(\underline{a}_i), \phi(\underline{a}_j) \in C_1$, it follows from (2) that the vector $\phi(\underline{x})$ is at distance at most $(t-2)$ from two distinct codewords of C_1. As Hamming spheres of radius $(t-2)$ about the codewords of C_1 are disjoint, this is impossible. □

The set of control arrays $\mathcal{U}_2(t)$ may be now obtained as follows. Let $\mathcal{A} = \{A_0, A_1, \ldots, A_n\}$ and $\mathcal{B} = \{B_0, B_1, \ldots, B_m\}$ be two fixed sets of $m \times n$ binary arrays. The elements of $\mathcal{A}$ and $\mathcal{B}$, namely A_i and B_j are defined by:

A_i is the $m \times n$ array with each *column* equal to the vector $\underline{a}_i$;

B_j is the $m \times n$ array with each *row* equal to the vector $\underline{b}_j$.

The set of control arrays $\mathcal{U}_2(t)$ is obtained as a direct sum of the sets $\mathcal{A}$ and $\mathcal{B}$

$$\mathcal{U}_2(t) = \{ A \oplus B : A \in \mathcal{A}, B \in \mathcal{B} \} .$$

Note that adding an element of $\mathcal{B}$ to an arbitrary $m \times n$ array V is equivalent to complementing a certain set of columns of V. This does not affect the number of $0 \leftrightarrow 1$ transitions in the columns of V since a vector and its complement contain the same number of $0 \leftrightarrow 1$ transitions. Similarly, adding an element of $\mathcal{A}$ to V does not affect the number of $0 \leftrightarrow 1$ transitions in the rows of V. However, if $\underline{v}_1, \underline{v}_2, \ldots, \underline{v}_n$ is the set of *columns* of V, then the set of columns of $V \oplus A_i$ is just $\underline{v}_1 + \underline{a}_i, \underline{v}_2 + \underline{a}_i, \ldots, \underline{v}_n + \underline{a}_i$. In view of Proposition 1, for each specific column of V there is at most one element $A \in \mathcal{A}$, such that the number of $0 \leftrightarrow 1$ transitions in the corresponding column of $V \oplus A$ is less than t. Since the cardinality of $\mathcal{A}$ is strictly greater than the number of columns in V, there exists at least one $A^* \in \mathcal{A}$ such that *all* the columns of $V \oplus A^*$ contain at least t transitions. A similar argument shows that there exists at least one $B^* \in \mathcal{B}$ such that all the number of $0 \leftrightarrow 1$transitions in all the *rows* of $V \oplus B^*$ is at least t. Therefore $V \oplus U$, where $U = A^* \oplus B^* \in \mathcal{U}_2(t)$, satisfies property $P_2(t)$. Since the set of control arrays $\mathcal{U}_2(t)$ consists of $(m+1)(n+1)$ elements, in order to indicate which control array was applied to V we need exactly $\lceil \log_2(m+1)(n+1) \rceil$ redundant bits.

Example. Let $n = m = 3$, and $t = 2$. Since $t - 2 = 0$, any set of distinct vectors may be taken as C_1 and C_2. For instance we can employ $C_1 = C_2 = \{000, 011, 110, 101\}$. In this case $\{\underline{a}_0, \underline{a}_1, \underline{a}_2, \underline{a}_3\} = \{\underline{b}_0, \underline{b}_1, \underline{b}_2, \underline{b}_3\} = \{000, 001, 010, 011\}$. Hence we have:

$$A_0 = \begin{matrix} 0\,0\,0 \\ 0\,0\,0 \\ 0\,0\,0 \end{matrix} \qquad A_1 = \begin{matrix} 0\,0\,0 \\ 0\,0\,0 \\ 1\,1\,1 \end{matrix} \qquad A_2 = \begin{matrix} 0\,0\,0 \\ 1\,1\,1 \\ 0\,0\,0 \end{matrix} \qquad A_3 = \begin{matrix} 0\,0\,0 \\ 1\,1\,1 \\ 1\,1\,1 \end{matrix}$$

$$B_0 = \begin{matrix} 0\,0\,0 \\ 0\,0\,0 \\ 0\,0\,0 \end{matrix} \qquad B_1 = \begin{matrix} 0\,0\,1 \\ 0\,0\,1 \\ 0\,0\,1 \end{matrix} \qquad B_2 = \begin{matrix} 0\,1\,0 \\ 0\,1\,0 \\ 0\,1\,0 \end{matrix} \qquad B_3 = \begin{matrix} 0\,1\,1 \\ 0\,1\,1 \\ 0\,1\,1 \end{matrix}$$

The set of control arrays $\mathcal{U}_2(2)$ thus consists of the following 16 arrays:

$$\begin{matrix} 0\,0\,0 & 0\,0\,0 & 0\,0\,0 & 0\,0\,0 & 0\,0\,1 & 0\,0\,1 & 0\,0\,1 & 0\,0\,1 \\ 0\,0\,0 & 0\,0\,0 & 1\,1\,1 & 1\,1\,1 & 0\,0\,1 & 0\,0\,1 & 1\,1\,0 & 1\,1\,0 \\ 0\,0\,0 & 1\,1\,1 & 0\,0\,0 & 1\,1\,1 & 0\,0\,1 & 1\,1\,0 & 0\,0\,1 & 1\,1\,0 \end{matrix}$$

$$\begin{matrix} 0\,1\,0 & 0\,1\,0 & 0\,1\,0 & 0\,1\,0 & 0\,1\,1 & 0\,1\,1 & 0\,1\,1 & 0\,1\,1 \\ 0\,1\,0 & 0\,1\,0 & 1\,0\,1 & 1\,0\,1 & 0\,1\,1 & 0\,1\,1 & 1\,0\,0 & 1\,0\,0 \\ 0\,1\,0 & 1\,0\,1 & 0\,1\,0 & 1\,0\,1 & 0\,1\,1 & 1\,0\,0 & 0\,1\,1 & 1\,0\,0 \end{matrix}$$

□

In order to obtain an $m \times n$ array which simultaneously satisfies P_1 and $P_2(t)$ we proceed as follows. A set of control arrays $\mathcal{U}_2(t+2)$ is designed as described above for the property $P_2(t+2)$. An element $U \in \mathcal{U}_2(t+2)$ is applied to an arbitrary information array V to ensure that $V \oplus U$ satisfies $P_2(t+2)$. Then several bits of $V \oplus U$ are complemented in the horizontal readout order of (1) so that the resulting array, $V \oplus U \oplus W$ say, satisfies property P_1. Yet $V \oplus U \oplus W$ must also satisfy $P_2(t)$.

Proposition 2. *Let the vector* $\underline{x} = (x_1, x_2, \ldots, x_n) \in F_n$ *have* t *transitions of the type* $0 \leftrightarrow 1$. *If* $\underline{x}' = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_i, x_{i+1}, x_{i+2}, \ldots, x_n)$, *where* $\bar{a}$ *denotes the binary complement of* a, *then the number of* $0 \leftrightarrow 1$ *transitions in* $\underline{x}'$ *is at least* $t - 2$.

Proof. Clearly the only $0 \leftrightarrow 1$ transitions which could be present in $\underline{x}$ but not in $\underline{x}'$ are from x_i to x_{i+1} or from x_n to x_1. Thus complementing the first i bits of $\underline{x}$ could eliminate at most two $0 \leftrightarrow 1$ transitions. □

Since the bits of $V \oplus U$ are complemented in the horizontal readout order, each row (column) of $V \oplus U \oplus W$ complies with the conditions of Proposition 2 with respect to the corresponding row (column) of $V \oplus U$. As $V \oplus U$ satisfies $P_2(t+2)$, it follows that $V \oplus U \oplus W$ satisfies both P_1 and $P_2(t)$. Notably, the whole process does not require a conversion table. The only information that has to be stored in the encoder is the two sets of vectors $\underline{a}_0, \underline{a}_1, \ldots, \underline{a}_n$ and $\underline{b}_0, \underline{b}_1, \ldots, \underline{b}_m$ which define the control set $\mathcal{U}_2(t)$. The control array $U \in \mathcal{U}_2(t)$, which is applied to V so that $V \oplus U$ satisfies $P_2(t+2)$, may be determined by simply complementing the rows, and subsequently the columns, of V. Then the control array W can be determined by complementing the bits of $V \oplus U$ in the order defined by (1). The total number of redundant bits required by the encoder is of the order of $2 \log_2(mn)$.

We now consider the outer error-correcting encoder. Indeed, the outer code should properly exploit the properties of the inner code. Since the errors in the channel are of asymmetric type, while the transmitted symbols (the $m \times n$ arrays) are balanced — any number of errors in the array will cause an imbalance. Thus, a symbol in error will be detected, resulting in a symbol erasure. In the language of coding, such an event is called an erasure. Of course, it is much easier to correct erasures than errors. For this purpose we can use either the Reed Solomon codes, or the codes described in [2, 3, 4] when larger symbols are involved.

3. An implementation example

Assume that the information is arranged in 16×16 arrays, hereafter referred to as symbols. First, an outer code can be applied. For instance we may have an array of 16×16 symbols, where some 254 symbols carry information, while the remaining 2 symbols are used for redundancy. We can also employ two-dimensional interleaving as described in [1]. After encoding all the symbols using the outer code, the inner modulation code is applied to each 16×16 symbol. The following set of 17 control vectors of length 16 is employed to ensure that the 16×16 array satisfies $P_2(4)$:

$$\begin{array}{llll} \underline{a}_0 = \underline{b}_0 = & 0000000000000000 & \underline{a}_9 = \underline{b}_9 = & 0001001101011111 \\ \underline{a}_1 = \underline{b}_1 = & 0000110100000111 & \underline{a}_{10} = \underline{b}_{10} = & 0110011100100000 \\ \underline{a}_2 = \underline{b}_2 = & 0111100101111000 & \underline{a}_{11} = \underline{b}_{11} = & 0110101000100111 \\ \underline{a}_3 = \underline{b}_3 = & 0111010001111111 & \underline{a}_{12} = \underline{b}_{12} = & 0010001011100000 \\ \underline{a}_4 = \underline{b}_4 = & 0011110010111000 & \underline{a}_{13} = \underline{b}_{13} = & 0010111111100111 \\ \underline{a}_5 = \underline{b}_5 = & 0011000110111111 & \underline{a}_{14} = \underline{b}_{14} = & 0101101110011000 \\ \underline{a}_6 = \underline{b}_6 = & 0100010111000000 & \underline{a}_{15} = \underline{b}_{15} = & 0101011010011111 \\ \underline{a}_7 = \underline{b}_7 = & 0100100011000111 & \underline{a}_{16} = \underline{b}_{16} = & 0000111100101000 \\ \underline{a}_8 = \underline{b}_8 = & 0001111001011000 & & \end{array}$$

It is easy to verify that the set $\{\phi(\underline{a}_i) : 0 \leq i \leq 16\}$ is a two-error-correcting code with all the codewords of even weight. Hence, according to Proposition 1, given any 16×16 array we can always choose an $\underline{a}_i$, such that adding $\underline{a}_i$ to all the columns in the array yields at least 4 transitions in each column. Similarly, we can always choose a $\underline{b}_j$, such that after adding $\underline{b}_j$ to all the rows in the array we have at least 4 transitions per row.

We now establish a 1-1 relationship between the indices i and balanced vectors of length 6. For instance, these vectors may ordered lexicographically as in the following table.

Table 1

0	000111	6	010110	12	100110
1	001011	7	011001	13	101001
2	001101	8	011010	14	101010
3	001110	9	011100	15	101100
4	010011	10	100011	16	110001
5	010101	11	100101		

In this example we encode each 16×16 array X into a 17×17 array as follows:

1. Find the first i such that adding $\underline{a}_i$ to each column of X produces at least 4 transitions in each column, and the first j such that adding $\underline{b}_j$ to each row yields at least 4 transitions in each row. Add $\underline{a}_i$ to each column and $\underline{b}_j$ to each row.

2. Append to the array obtained so far a row whose entries are specified as follows. The first 6 bits consist of the vector corresponding to i in Table 1. The next 6 bits are the vector corresponding to j in Table 1. Finally, the last 4 bits are arbitrary — for instance use the vector 1010. This produces a 17×16 array.

3. Consider the first 16 rows of the array obtained so far. Let l be the number of bits that need be complemented to balance these 16 rows according to the procedure of Henry-Knuth [5, 6]. Notice that $0 \leq l \leq 255$ and hence we need 8 bits, say $a_7a_6a_5a_4a_3a_2a_1a_0$, to write l in binary form. Then $a_7a_6a_5a_4$ are taken to be the binary form of an integer l_1 and $a_3a_2a_1a_0$ are taken to be the binary form of an integer l_2, where $l_1, l_2 \leq 15$. We associate l_1 and l_2 with balanced vectors according to Table 1. This produces a vector of length 12, that may be concatenated with 10101 say, to obtain a vector of length 17 which is appended as the last column of the array.

The demodulation process is just the inverse of the modulation process. That is, the three steps above are performed in the reversed order. However, prior to the actual demodulation each symbol is checked to see whether it contains exactly 145 zeros. If not, and this will always be the case when asymmetric errors occur, the demodulator declares an erasure at this symbol and passes the information to the outer error-correcting code. Thus all asymmetric errors are detected.

Example. Assume that we wish to modulate the 16×16 identity array X with 1's on its main diagonal only. There are exactly two (cyclic) transitions in each row and column. Adding the vector $\underline{a}_1$ to each row and to each column increases the number of transitions to at least 4. Let X' be the array obtained this way:

$$X' = \begin{pmatrix}
1&0&0&0&1&1&0&1&0&0&0&0&0&1&1&1\\
0&1&0&0&1&1&0&1&0&0&0&0&0&1&1&1\\
0&0&1&0&1&1&0&1&0&0&0&0&0&1&1&1\\
0&0&0&1&1&1&0&1&0&0&0&0&0&1&1&1\\
1&1&1&1&1&0&1&0&1&1&1&1&1&1&0&0&0\\
1&1&1&1&0&1&1&0&1&1&1&1&1&1&0&0&0\\
0&0&0&0&1&1&1&1&0&0&0&0&0&1&1&1\\
1&1&1&1&0&0&1&1&1&1&1&1&1&1&0&0&0\\
0&0&0&0&1&1&0&1&1&0&0&0&0&1&1&1\\
0&0&0&0&1&1&0&1&0&1&0&0&0&1&1&1\\
0&0&0&0&1&1&0&1&0&0&1&0&0&1&1&1\\
0&0&0&0&1&1&0&1&0&0&0&1&0&1&1&1\\
0&0&0&0&1&1&0&1&0&0&0&0&1&1&1&1\\
1&1&1&1&0&0&1&0&1&1&1&1&1&1&0&0\\
1&1&1&1&0&0&1&0&1&1&1&1&1&1&0&1&0\\
1&1&1&1&0&0&1&0&1&1&1&1&1&1&0&0&1
\end{pmatrix}$$

According Table 1 the index of $\underline{a}_1$ corresponds to the vector 001011. Hence the last row of the modulated array should be 0010110010111010, produces a 17×16 array. We now need to balance this array. The first matrix W balancing X' has 122 ones. Adding W to X' – or complementing the first 122 bits of X' – we obtain

$$X'' = \begin{pmatrix}
0&1&1&1&0&0&1&0&1&1&1&1&1&0&0&0\\
1&0&1&1&0&0&1&0&1&1&1&1&1&0&0&0\\
1&1&0&1&0&0&1&0&1&1&1&1&1&0&0&0\\
1&1&1&0&0&0&1&0&1&1&1&1&1&0&0&0\\
0&0&0&0&0&1&0&1&0&0&0&0&0&1&1&1\\
0&0&0&0&1&0&0&1&0&0&0&0&0&1&1&1\\
1&1&1&1&0&0&0&0&1&1&1&1&1&0&0&0\\
0&0&0&0&1&1&0&0&0&0&1&1&1&0&0&0\\
0&0&0&0&1&1&0&1&1&0&0&0&0&1&1&1\\
0&0&0&0&1&1&0&1&0&1&0&0&0&1&1&1\\
0&0&0&0&1&1&0&1&0&0&1&0&0&1&1&1\\
0&0&0&0&1&1&0&1&0&0&0&1&0&1&1&1\\
0&0&0&0&1&1&0&1&0&0&0&0&1&1&1&1\\
1&1&1&1&0&0&1&0&1&1&1&1&1&1&0&0\\
1&1&1&1&0&0&1&0&1&1&1&1&1&1&0&1&0\\
1&1&1&1&0&0&1&0&1&1&1&1&1&1&0&0&1
\end{pmatrix}$$

Finally, we have to generate the last column of the array. The binary expansion of 122 is 01111010. We break these 8 bits into two 4-bit vectors: 0111 which is the binary representation of 7, and 1010 which is the binary representation of 10. Again referring to Table 1, we find that 7 corresponds to the vector 011001 and 10 corresponds to the vector 100011. Thus, the last column of the array is 01100110001110101. The final modulated 17×17 array is given by:

$$Y = \begin{pmatrix}
0&1&1&1&0&0&1&0&1&1&1&1&1&0&0&0&0\\
1&0&1&1&0&0&1&0&1&1&1&1&1&0&0&0&1\\
1&1&0&1&0&0&1&0&1&1&1&1&1&0&0&0&1\\
1&1&1&0&0&0&1&0&1&1&1&1&1&0&0&0&0\\
0&0&0&0&0&1&0&1&0&0&0&0&0&1&1&1&0\\
0&0&0&0&1&0&0&1&0&0&0&0&0&1&1&1&1\\
1&1&1&1&0&0&0&0&1&1&1&1&1&0&0&0&1\\
0&0&0&0&1&1&0&0&0&0&1&1&1&0&0&0&0\\
0&0&0&0&1&1&0&1&1&0&0&0&0&1&1&1&0\\
0&0&0&0&1&1&0&1&0&1&0&0&0&1&1&1&0\\
0&0&0&0&1&1&0&1&0&0&1&0&0&1&1&1&1\\
0&0&0&0&1&1&0&1&0&0&0&1&0&1&1&1&1\\
0&0&0&0&1&1&0&1&0&0&0&0&1&1&1&1&1\\
1&1&1&1&0&0&1&0&1&1&1&1&1&1&0&0&0\\
1&1&1&1&0&0&1&0&1&1&1&1&1&0&1&0&1\\
1&1&1&1&0&0&1&0&1&1&1&1&1&0&0&1&0\\
0&0&1&0&1&1&0&0&1&0&1&1&1&0&1&0&1
\end{pmatrix}$$

□

In addition to the foregoing procedure, there are of course many other possibilities to implement the proposed encoder. For instance, if the channel is dominated by asymmetric errors, the most attractive possibility would be to employ the error-correcting code as an inner code and the modulation encoder as an outer code. This rather unconventional scheme is appropriate in our situation due to the fact that the error-correcting code is systematic and features a very high rate. For instance, we may consider 254 information arrays of size 15×15 and modulate them into 16×16 symbols, such that each 16×16 symbol is balanced (i.e., has exactly 128 zeros) and contains at least two transitions in each row and column. The required modulation procedure is completely analogous to the one described above. Two or more symbols, that can be placed anywhere in the 16×16 array of symbols, are used for redundancy and are encoded using two-dimensional interleaving, as described in [1]. These redundant symbols would not necessarily satisfy the modulation requirements, but since there are very few of them, they cannot create a gross imbalance in the whole array. Moreover, if asymmetric errors occur which affect one ore more of the information symbols, they will be detected as erasures. Failures in the redundancy symbols cannot be detected directly, except by the error-correcting code.

References

[1] M. Blaum and J. Bruck, "Correcting two-dimensional clusters by interleaving of symbols," *IEEE Int. Symp. Inform. Theory*, Trondheim, Norway, 1994.

[2] M. Blaum, J. Bruck, and A. Vardy, "MDS array codes with independent parity symbols," IBM Research Report, RJ 9883, September 1994.

[3] M. Blaum, H. Hao, R. Mattson and J. Menon, "A Coding Technique for Double Disk Failures in Disk Arrays," US Patent 5,271,012, 1993.

[4] M. Blaum and R.M. Roth, "New array codes for multiple phased burst correction," *IEEE Trans. Inform. Theory*, vol. 39, pp. 66–77, 1993.

[5] P.S. Henry, "Zero disparity coding system," US Patent 4,309,694, 1982.

[6] D.E. Knuth, "Efficient balanced codes," *IEEE Trans. Inform. Theory*, vol. 32, pp. 51–53, 1986.

[7] S.A. Lis and P.D. Henshaw, "Ultra-Dense Optical Mass Storage," AD-A232 767, report prepared for USAF, AFSC, February 1991.

[8] F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error-Correcting Codes*, New York: North-Holland, 1977.

Homogeneous Trellis Codes

Mitchell D. Trott and James P. Sarvis

Laboratory for Information and Decision Systems
MIT 35-213
Cambridge, MA 02139

trott@lids.mit.edu, jpsarvis@lids.mit.edu

Abstract. A trellis code is *homogeneous* if it can be represented as the orbit $\mathcal{C} = C\boldsymbol{x}$ of a seed sequence $\boldsymbol{x}$ under the action of a time-invariant group system C over a permutation alphabet. Most (but not all) useful trellis codes are homogeneous; in particular, all geometrically uniform trellis codes are homogeneous, as are all trellis codes constructed as a cascade of a linear convolutional or group code and a one-to-one map from group elements to signal points.

We show how to test if a trellis code is homogeneous. The test uses standard algorithms from computational group theory to construct the symmetry group of the code. It turns out that some published *nonlinear* trellis codes are homogeneous, though most are not. We then prove that there exists no homogeneous version of the 64-state nonlinear trellis code adopted in the recent V.34 modem standard.

1 Introduction

Most useful trellis codes are based on binary linear convolutional codes. The algebraic theory available for binary linear systems greatly simplifies the problem of constructing new codes of this type. However, codes designed for rotational invariance are often nonlinear, and must be constructed by *ad hoc* methods. A less structured algebraic class may provide the needed algebraic setting for some of these nonlinear codes. The class of group codes is one possible candidate [4, 7]. It remains to see, however, how many nonlinear codes can be represented using groups.

In this paper we develop a method for testing if a particular trellis code has a description as a group code, and apply this test to the nonlinear codes in the literature. We then prove that no code with the parameters of the one selected for the V.34 standard has a group code description.

The paper is organized as follows. In Section 2 we define the objects used in the remainder of the paper, including Latin codes and derivative trellis sections. Both concepts are used heavily in later sections. In Section 3 we show how to convert the problem of finding the symmetry group of a code into one of finding the symmetry trellis. In Section 4 we describe how to compute the symmetry trellis, and list the results of this algorithm applied to various nonlinear trellis codes in the literature. In Section 5 we prove that there is no homogeneous equivalent of the V.34 code. The proof is rather involved and does not easily generalize; however, most of the ideas used in the proof can be applied in a suitably weakened form to larger codes.

2 Definitions

A *trellis code* $\mathcal{C}$ is the set of bi-infinite label sequences generated by walks on a finite, labeled, directed graph. The label alphabet is either a finite signal set A in N-dimensional Euclidean space, or a set A/A' of cells of a partitioned signal set [3, 12]. Except for a few places in Section 5, we shall not be greatly interested in the structure of the signal set in Euclidean space. For our purposes, a trellis code is essentially a *graph shift* [6]. We always assume that A contains a distinguished "zero" element 0, and that $\mathcal{C}$ contains the constant zero sequence $\mathbf{0}$.

A (time-invariant) *group code* C over a group G is a closed, shift-invariant subgroup of $G^{\mathbf{Z}}$. In this paper, G will usually be the group of all permutations of the signal set A. The identity of G will be denoted by id, and the group operation by juxtaposition. The italic letter C will be used for group codes and the script letter $\mathcal{C}$ will be for codes over a non-group alphabet.

The *symmetry group* of a code $\mathcal{C}$ is the set of all permutation sequences $\boldsymbol{c} \in G^{\mathbf{Z}}$ such that $\boldsymbol{c}\mathcal{C} = \mathcal{C}$. The symmetry group is in fact a group code, the *symmetry code* of C. A code $\mathcal{C}$ is *homogeneous* if its symmetry group acts transitively, *i.e.*, if for any two code sequences $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ in $\mathcal{C}$, there exists a symmetry $\boldsymbol{c} \in C$ such that $\boldsymbol{c}\boldsymbol{x}_1 = \boldsymbol{x}_2$. A code is homogeneous if and only if it is the orbit $C(\mathbf{0})$ of the zero sequence under the action of a group code C [10]. When the alphabet G is a group of isometries, the orbit $C(0)$ is a *geometrically uniform* code [3].

As shown in [4], every group code C has a unique minimal trellis section $T \subseteq \Sigma \times G \times \Sigma$, where $\Sigma = C/C\_C_+$ is the state group of C. The elements of T are *branches* $b = (s, l, s')$. Every code sequence $\boldsymbol{c} \in C$ passes through a well-defined state $\sigma_k(c)$ at each time k, where $\sigma_k : C \to \Sigma$ is the natural homomorphism. The trellis section T is unique and minimal in the sense that any other trellis section for C can be reduced to T by merging states.

A trellis code $\mathcal{C}$ does not always have a unique minimal trellis section. However, we shall restrict our attention to a special subclass of trellis codes which do. A code $\mathcal{C}$ is a *Latin trellis code* if:

1. $\mathcal{C}$ has a unique minimal trellis section $\mathcal{T}$ in which no two branches leaving the same state have the same label, no two branches entering the same state have the same label, and each sequence of $\mathcal{C}$ appears along a unique path through $\mathcal{T}$.

2. The unlabeled trellis section $\mathcal{T}$ has the structure of a shift register graph, so that it "looks like" the state transition diagram of a feedforward linear encoder. All states have the same number f of entering and exiting branches, and there are no parallel branches.

3. The signal set A is partitioned into m *colors*, each of size f. The f labels on branches leaving each state of $\mathcal{T}$ all have the same color, as do the f branches entering each state.

Combined, these requirements ensure that the trellis section $\mathcal{T}$ may be partitioned into fully connected subtrellises $\mathcal{T}_1, \ldots, \mathcal{T}_n$ such that all branches in a subtrellis have the same

color. The labeling of a fully-connected subtrellis $\mathcal{T}_i$ is described by an $f \times f$ matrix $\mathcal{L}_i$ which indicates a branch $b = (s, l, s')$ by an entry of l at the intersection of row s and column s'. The matrices $\{\mathcal{L}_i\}$ are Latin squares, hence the name "Latin code." With the exception of a few published nonlinear rotationally invariant codes, all useful trellis codes designed for additive white Gaussian noise channels are Latin. Empirical evidence suggests that non-Latin codes have inferior Euclidean distance profiles to Latin codes with similar complexity.

The trellis section of a Latin code always has exactly f self-transition branches (s, l, s) which correspond to the f constant sequences in $\mathcal{C}$. We identify the "zero" state 0 as the unique state that begins and ends the zero-labeled branch $(0, 0, 0)$.

There is a considerably literature on Latin squares and their symmetry properties [5]. It is not hard to show, for example, that all Latin squares $\mathcal{L}_i$ that occur in the trellis of a homogeneous Latin code must be isotopic, *i.e.*, equivalent up to relabeling. The Latin squares need not be group tables, though for all published homogeneous codes they are always the group table of $(Z_2)^n$. This explains to some extent the intuitive perception that many nonlinear codes are "nearly linear."

The concepts defined next apply only to codes with unique minimal trellis sections (such as Latin codes and group codes). A code $\mathcal{C}$ is *k-controllable* if any state in $\mathcal{T}$ can be connected to any other with a path of length k or less. The *k-reachable state set* of $\mathcal{C}$ is the set of states that may be reached from the zero state in k transitions. The *k-drivable state set* of $\mathcal{C}$ is the set of states that reach the zero state in k transitions. The *k-controllable state set* of $\mathcal{C}$ is the intersection of the k-reachable and k-drivable state sets; the *k-controllable subcode* of $\mathcal{C}$ is the set of sequences that pass exclusively through the k-controllable states. For a group code C, the k-controllable state set and k-controllable subcode are groups.

The *derivative code* $\mathcal{C}'$ of $\mathcal{C}$ is the set of all state sequences traversed by code sequences in $\mathcal{C}$ [7, 11]. The derivative of a Latin code is always Latin. The *derivative trellis* $\mathcal{T}'$ of $\mathcal{T}$ is the trellis section of the derivative code. Its state set is termed the *derivative state set*, its label alphabet is the state set of $\mathcal{C}$, and each label appears on exactly one branch. If (a, s, a') is the branch in the derivative trellis that corresponds to state s, then a is the *left derivative* of s, and a' is the *right derivative* of s. We will refer to states of $\mathcal{C}$ and branches of $\mathcal{T}'$ interchangeably, for labeled paths through $\mathcal{T}'$ uniquely correspond to state sequences through $\mathcal{T}$.

The derivative state set has n elements, where n is the number of fully-connected subtrellises in $\mathcal{T}$. Every pair of derivative states is connected by l parallel branches in the derivative trellis, where l is the number of states in the 1-controllable subcode of $\mathcal{C}$. If $\mathcal{C}$ is k-controllable then $\mathcal{C}'$ is $k - 1$ controllable.

Branches in a Latin trellis are colored by their labels. The left state s of a branch (s, l, s') takes the color of its exiting branches; the right state s' takes the color of its entering branches. The left and right colors of a state may differ. Derivative states may also be assigned colors. A left derivative state takes the right-sided color of the states that leave it. A right derivative state takes the left-sided color of the states that enter it. With these definitions, the left derivative of s determines the color of the branches entering s, while the right derivative of s determines the color of the branches leaving s.

Example 1: The V.34 code has a trellis section with 64 states and $f = 16$ branches leaving each state. The label alphabet has 32 elements and is partitioned into two colors of 16 elements each. We will refer to the color containing the label 0 as the "even" color, and the other as the "odd" color. (In general, the label alphabet of a rate $k/k+j$ code is partitioned into 2^j colors.) There are 4 fully connected subtrellises, each with 16 states. The 1-controllable subcode has 4 states. The derivative trellis has 4 states, is fully connected, and each pair of derivative state is connected by 4 parallel branches. ■

3 The basic result

Our basic result is that all information about the action of the symmetry code C on $\mathcal{C}$ is contained in the action of the symmetry trellis T on $\mathcal{T}$. The problem of finding the infinite symmetry group C is therefore reduced to the problem of computing the finite symmetry trellis T.

To prove this, we need only show that symmetry states act on symbol states, *i.e.*, that the states of the symmetry trellis T are permutations of states of the symbol trellis $\mathcal{T}$. Then T may be viewed as a subgroup of $S_\Sigma \times S_A \times S_\Sigma$, where Σ and A are the state and label sets of $\mathcal{T}$, respectively, and where S_X denotes the group of all permutations of the set X.

Theorem 1 *The action of C on $\mathcal{C}$ induces an action of T on $\mathcal{T}$.*

Proof: Our assumptions guarantee that distinct states in $\mathcal{T}$ have disjoint sets of future half-infinite sequences, and disjoint sets of past half-infinite sequences. Thus if two permutation sequences agree either on the past or on the future, then they must have the same permutation action of the states of $\mathcal{T}$. This further implies that all sequences that pass through the same state in T have the same action on $\mathcal{T}$.

Finally, if two states s_1 and s_2 of T have the same action on the states of $\mathcal{T}$, then it can be shown that concatenating the pasts of s_1 to the futures of s_2 yields new symmetries of $\mathcal{C}$ that were not present in T, an impossibility. Thus distinct states of T have distinct permutation actions. ■

It can similarly be shown that for a homogeneous code, the symmetry group of the derivative code $\mathcal{C}'$ is the derivative code of the symmetry group C'.

4 Computing the symmetry group

The algorithm given below can be used to compute the symmetry group of any code with a unique minimal trellis section $\mathcal{T}$. Our goal is to find the *symmetry trellis* of $\mathcal{T}$, namely the group of all branches $b = (s, p, s')$ such that $b\mathcal{T} = \mathcal{T}$, where s and s' are permutations of the state set and p is a permutation of the label alphabet of $\mathcal{T}$. The symmetry group

C of $\mathcal{C}$ is the set of permutation sequences along bi-infinite paths through the symmetry trellis.

Most branches in the symmetry trellis are not part of bi-infinite walks. They may connect to branches on the left and right for a few transitions, but eventually reach a dead-end state with no exiting branches. These "stub" paths represent symmetries that are locally good but not extendible to bi-infinite symmetries of C. To be more specific, a stub symmetry of length l is a symmetry of the block code formed by restricting $\mathcal{C}$ to the finite time interval $[1, l]$. Given the symmetry trellis, then, a significant computational problem is finding the subtrellis T of branches that are part of bi-infinite paths.

The algorithm first computes the symmetry trellis, then deletes the branches not part of bi-infinite walks. The necessary computations are readily performed using a computational group theory package such as Gap or Magma. Unfortunately, these packages do not understand directed labeled graphs, so a standard trick in graph theory must be used to convert a labeled, directed graph $\mathcal{T}$ into an undirected, unlabeled graph $\mathcal{V}$ with the same symmetry properties.

Define $\mathcal{V}$ to have one state for each branch of $\mathcal{T}$. Two states b_1 and b_2 of $\mathcal{V}$ are connected by an unlabeled edge if, as branches in $\mathcal{T}$, they leave the same state, they enter the same state, or if they have the same label. Then the graph automorphism group of $\mathcal{V}$ is essentially the symmetry trellis of $\mathcal{T}$. Graph automorphisms are computed automatically by Gap or Magma.

To following iterative algorithm finds the branches that are part of bi-infinite walks on the symmetry trellis:

1. Let $i = 1$, and let B_i be the group of branches in the symmetry trellis.
2. Intersect the left states of B_i with the right states of B_i to form S_i, the group of states that are part of symmetry paths of length $2i$.
3. Let B_{i+1} be the subgroup of branches in B_i that connect states in S_i to states in S_i. Increment i, and return to step 2 until a fixed point is reached.

The final set B_i is the symmetry trellis T. If it acts transitively on $\mathcal{T}$ then the code $\mathcal{C}$ is homogeneous.

4.1 Symmetries of various nonlinear codes

Using the algorithm developed above, we have tested the nonlinear codes of Wei [13], Pietrobon *et al.* [8, 9], Calderbank [1], Chouly and Sari [2], and the nonlinear codes in the V.34 and V.32 standards for homogeneity. Of these codes, the 8-state V.32 code is homogeneous with a state group of D_4 and the 16-state rate-3/4 codes in [13] and [2] are homogeneous with a state group of $D_4 \times \mathbf{Z}_2$. Most of the Pietrobon *et al.* codes can be shown to be nonhomogeneous by inspection, for their encoders have fewer constant input sequences than constant output sequences. This is not possible for a homogeneous code [10]. The practical complexity limit for direct computation of the trellis symmetry group appears to be about 256 states.

5 Nonexistence of a homogeneous V.34-like code

In this section we show that there is no homogeneous or geometrically uniform equivalent of the 64 state rotationally invariant code selected for the ITU V.34 modem standard.

Let $\mathcal{C}$ be a homogeneous 2-controllable 64-state rate-4/5 Latin trellis code. We first show that $\mathcal{C}$ has no nontrivial automorphisms. We then show that if the symmetry trellis T has more than 64 states then the symmetry code C must contain nontrivial automorphisms, a contradiction. Finally, we show that for the partition used in the V.34 code, 90° rotational invariance and a 64-state symmetry group are incompatible.

5.1 The automorphism group

An *automorphism* of $\mathcal{C}$ is a symmetry sequence $\boldsymbol{c} \in C$ that fixes the zero sequence $\mathbf{0}$. We will show that $\mathcal{C}$ has no nontrivial automorphisms.

Lemma 2 *The 1-controllable subtrellis of a homogeneous V.34-like code has exactly two zero-labeled branches.*

Proof: The 1-controllable subcode has four states and is fully connected. It always include the zero branch $(0,0,0)$. By homogeneity, if it had three zero-labeled branches it would have four, which would make it catastrophic. Thus it has either one or two. To guarantee a good minimum distance and to ensure rotational invariance, we will show that some label must appear twice in the 1-controllable subtrellis. Homogeneity then forces every label that appears once to appear twice, including the zero label.

The minimum distance of the code is 5. Within the even-colored cells the partition, cell 0 has three neighbors at distance 4 and twelve at distance 2. The 1-controllable subtrellis contains three paths of length 2 that begin and end at the zero state. They must all have distance 6 from $\mathbf{0}$. The three cells at distance 4 from cell 0 must therefore be assigned to branches along these paths. As shown later, the 1-controllable subcode is 180° rotationally invariant. The rotation $(r^2, \rho^2(0), r^2)$ of the zero branch $(0,0,0)$ is therefore another self-transition branch in the 1-controllable subcode. Its label $\rho^2(0)$ has distance 4 from cell 0, and therefore appears twice in the 1-controllable subtrellis. ∎

A connected sequence $b_1, \ldots, b_n$ of zero-labeled branches in the orbit trellis $\mathcal{T}$ is a *maximal run of zeros* if b_1 cannot be preceded nor b_n followed by another zero-labeled branch. An automorphism of $\mathcal{C}$ must map a maximal run of zeros to another maximal run of zeros of the same length. This requirement sharply limits the size of the automorphism group of any rate $k/k+1$ code; for the V.34 code, it reduces it to the identity.

Lemma 3 *The automorphism group of a homogeneous V.34-like code is trivial.*

Proof: Any homogeneous 64-state rate-4/5 Latin trellis code has a unique maximal run of zeros of length 6. All branches and states along this path are therefore fixed by all automorphisms.

If state s begins a maximal run of zeros then its left derivative must be odd-colored. Similarly, if state s'' ends a maximal run of zeros, its right derivative must be odd-colored. Finally, if s' is in the middle of a zero run, its left and right derivative states must be even-colored. A maximal run of zeros can't remain in the same derivative state for more than three transitions unless the 1-controllable subcode of $\mathcal{C}$ has a pair of connected zero-labeled branches (which it doesn't). Thus, the length-6 run of zeros must pass through at least one of the two odd-colored derivative states and both of the even-colored derivative states. All derivative states in $\mathcal{T}'$ are therefore fixed by automorphism.

Since the automorphisms of $\mathcal{C}$ are contained in the 1-controllable subcode of C, they may be viewed as automorphisms of the 1-controllable subcode of $\mathcal{C}$. But the 1-controllable subcode of $\mathcal{C}$ has four states, two zero-labeled branches, and no parallel branches, hence it has a trivial automorphism group. The orbit code $\mathcal{C}$ therefore has no nontrivial automorphisms. ∎

5.2 Minimality of the orbit trellis

In general, the trellis T of a group code C can be much larger than the trellis $\mathcal{T}$ of the homogeneous code $\mathcal{C} = C(\mathbf{0})$ generated by C. We show that this is not possible for a code similar to the one in the V.34 standard.

Each branch in the symmetry trellis T is labeled with a permutation p. To find the minimal trellis $\mathcal{T}$ of the orbit code, each permutation is replaced with its action $p(0)$ on the cell 0 to form the trellis $T(0)$. The trellis $T(0)$ will be minimal if the following three conditions are satisfied: (i) all branches leaving the a state have distinct labels, (ii) all branches entering a state have distinct labels, and (iii) each bi-infinite code sequence appears on only one trellis path. If nonminimal, states of $T(0)$ are merged to arrive at the minimal trellis $\mathcal{T}$.

Lemma 4 *For a V.34-like code, the symmetry trellis T and the orbit trellis $\mathcal{T}$ have the same number of states.*

Proof: We first verify condition (iii). Assume that a code sequence appears on two distinct paths through $T(0)$. Then by homogeneity the zero sequence $\mathbf{0}$ must appear on two distinct bi-infinite paths through $T(0)$. This implies that $\mathcal{C}$ has a nontrivial automorphism, a contradiction.

We next verify condition (i). Assume two branches leaving a state of $T(0)$ have the same label. Then there exists a nonidentity branch $b = (\mathrm{id}, z, s)$ in T such that the permutation label z fixes 0. Consider the action of b on $\mathcal{T}$. It acts as the identity on left states and it preserves 0, hence it fixes all zero-labeled branches in $\mathcal{T}$. Every state in the 1-reachable state set of $\mathcal{T}$ has an entering zero-labeled branch, hence s fixes all 1-reachable states. In the derivative trellis $\mathcal{T}'$, the 1-reachable state set corresponds to the set of branches that leave the zero derivative state. The symmetry state s fixes these branches, hence it fixes the right derivative states of these branches. But this includes all derivative states, since $\mathcal{T}'$ is 1-controllable. Thus the left and right derivative states of s are the identity, and the branch b is in the 1-controllable subtrellis of T.

Since b is 1-controllable, there is another branch $b' = (s, z', \mathrm{id})$ from s back to the identity. The permutation z' must also fix 0, for otherwise the sequence $\boldsymbol{c} = (\ldots, \mathrm{id}, z, z', \mathrm{id}, \ldots) \in C$ would map $\mathbf{0}$ to a sequence with a single nonzero component, and there are no such sequences in $\mathcal{C}$. The sequence $\boldsymbol{c}$ consists entirely of permutations that fix 0, and is therefore an automorphism of $\mathcal{C}$, a contradiction.

Finally, we verify condition (ii) using a time-reversal of the above argument. ∎

5.3 Rotational invariance vs. homogeneity

We first show that the "square" $\boldsymbol{c}^2 = \boldsymbol{cc}$ of any sequence $\boldsymbol{c} \in C$ is a member of the 1-controllable subcode. We do this by proving that every derivative state of C has order 2. Then, we show that rotational invariance implies the existence of connected pair of zero-labeled branches, both of order 4. Finally, we show that these two properties conflict.

The derivative state group of the symmetry trellis has four states, as does the 1-controllable subcode. These groups are either $\mathbf{Z}_4$ or $\mathbf{Z}_2 \times \mathbf{Z}_2$. Using the rotational invariance of $\mathcal{C}$, we will show that both are $\mathbf{Z}_2 \times \mathbf{Z}_2$.

Rotational invariance implies that the constant rotation sequence $\boldsymbol{\rho}$ is a symmetry of $\mathcal{C}$, where ρ is the permutation of partition cells induced by 90° rotation. The permutation action of $\boldsymbol{\rho}$ on states has order 4. If it had order 2, then 180° rotation would map the zero state to itself, which occurs only when the partition has 180° rotational transparency.

Lemma 5 *The* 180° *rotation sequence* $\boldsymbol{\rho}^2$ *is in the 1-controllable subcode of* C.

Proof: The 32-way partition used in the V.34 code has the special property that 90° rotation fixes colors. That is, the set of labels on branches that leave the zero state is invariant under rotation. (This is not true, for example, for the partition used in the V.32 code.) Rotation must also fix colors in the derivative trellis $\mathcal{T}'$. Since there are two derivative states of each color, the action of $\boldsymbol{\rho}$ on derivative states has order at most 2, which implies that $\boldsymbol{\rho}^2$ fixes all states in the derivative trellis. It is therefore in the 1-controllable subcode of C. ∎

Lemma 6 *The state group of the 1-controllable subcode of* C *is isomorphic to* $\mathbf{Z}_2 \times \mathbf{Z}_2$.

Proof: We have already shown that the 1-controllable subtrellis of $\mathcal{T}$ contains exactly one nonzero branch with a zero label. The 1-controllable subtrellis of T therefore contains a nonidentity branch $b = (s, z, s')$ with a zero-fixing permutation label z. This branch must have order 2, otherwise its square b^2 would be a second nonidentity branch with a zero-fixing permutation label. Since s and s' both have order 2, the state group of the 1-controllable subcode of C must be $\mathbf{Z}_2 \times \mathbf{Z}_2$. ∎

The next two lemmas give general results about the orders of elements in state groups.

Lemma 7 *Let T be a symmetry trellis that has the same size as the orbit trellis $\mathcal{T}$ it generates. If two states in T are connected by a branch with a zero-fixing permutation then both states have the same order.*

Proof: Let $b = (s, z, s')$ be a branch with a zero-fixing permutation label z, and let m and n be the orders of s and s', respectively, with $m \neq n$. The branch $b^{1+\min(m,n)}$ fixes the zero symbol yet either merges with or diverges from b. Identically labeled branches cannot enter or leave the same state, hence $m = n$. ∎

Lemma 8 *The order of a derivative state divides the least common multiple of the orders of the branches (code states) that enter and leave it.*

Proof: If $s^n = s$, and (a, s, b) is a branch in the derivative trellis, then $a^n = a$ and $b^n = b$ because the derivative trellis is 1-observable. ∎

Lemma 9 *The derivative state group of C is $\mathbf{Z}_2 \times \mathbf{Z}_2$.*

Proof: The zero-fixing branch in the 1-controllable subcode of C must be part of a maximal run of zeros which starts in a state s with odd-colored left derivative and ends in a state s' with odd-colored right derivative. All branches and states in this maximal run of zeros have order 2 (by Lemma 7), hence all derivative states along this run also have order 2 (by Lemma 8). We have already shown that the derivative of r, an even-colored derivative state, has order 2, hence both an even- and odd-colored derivative state have order 2. Thus the derivative state group is $\mathbf{Z}_2 \times \mathbf{Z}_2$. ∎

Every element of the derivative state group has order 2, hence the square s^2 of any state s must have identity left and right derivative state. This implies that $\boldsymbol{c}^2$ is in the 1-controllable subcode of C for all $\boldsymbol{c} \in C$.

Finally, let r be the state in T passed through by the 90° rotation sequence $\boldsymbol{\rho}$. We have already shown that r has order 4 and that the derivative of r has order 2. Since r is even-colored (on both the left and right), a branch with a zero-fixing permutation enters r and another leaves r. The squares of these branches are distinct nonidentity branches in the 1-controllable subtrellis of T, both labeled with permutations that fix 0. Yet the 1-controllable subtrellis of $\mathcal{T}$ has only one nonzero zero-labeled branch, which proves that the desired code does not exist.

6 Conclusion

It is not particularly disappointing that many existing nonlinear codes fail to be homogeneous, as they are generally inferior to their linear, non-rotationally invariant counterparts. But it is unfortunate that the 64-state V.34 code is "inherently" nonuniform. The next logical step is to apply the methods developed here to construct a 256 state homogeneous code in the V.34 family, or to prove that none exists.

References

[1] A. R. Calderbank, "The differential encoding of coset codes by algebraic methods," in *Coding thoery and design theory, Part I: Coding theory* (D. Ray-Chaudhuri, ed.), pp. 16–34, Springer-Verlag, 1990.

[2] A. Chouly and H. Sari, "Six-dimensional trellis-coding with QAM signal sets," *IEEE Trans. Commun.*, vol. 40, no. 1, pp. 24–33, 1992.

[3] G. D. Forney, Jr., "Geometrically uniform codes," *IEEE Trans. Inform. Theory*, vol. IT-37, pp. 1241–1260, Sept. 1991.

[4] G. D. Forney, Jr. and M. D. Trott, "The dynamics of group codes: State spaces, trellis diagrams and canonical encoders," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1491–1513, Sept. 1993.

[5] D. Jungnickel, "Latin squares, their geometries and their groups. a survey," in *Coding Theory and Design Theory, Part II* (D. Ray-Chaudhuri, ed.), vol. 21 of *IMA Volumes in Mathematics and its Applications*, pp. 166–225, Springer, 1992.

[6] D. Lind and B. Marcus, "An introduction to symbolic dynamics." Preprint, 1994.

[7] H.-A. Loeliger and T. Mittelholzer, "Convolutional codes over groups," submitted to *IEEE Trans. Inform. Theory.*

[8] S. S. Pietrobon, D. J. Costello, Jr., and G. Ungerboeck, "Rotationally invariant trellis codes," in *Coded Modulation and Bandwidth-Efficient Transmission* (E. Biglieri and M. Luise, eds.), pp. 55–65, Elsevier Science Publishers B. V., 1992.

[9] S. S. Pietrobon, G. Ungerboeck, and D. J. Costello, Jr., "Rotationally invariant nonlinear trellis codes for two-dimensional modulation," submitted to *IEEE Trans. Inform. Theory.*

[10] N. T. Sindhushayana. PhD thesis, Cornell, 1994.

[11] M. D. Trott, *The Algebraic Structure of Trellis Codes.* PhD thesis, Stanford University, Aug. 1992.

[12] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. Inform. Theory*, vol. IT-28, no. 1, pp. 55–67, 1982.

[13] L. F. Wei, "Rotationally invariant trellis-coded modulations with multidimensional m-PSK," *IEEE J. Select. Areas Commun.*, vol. SAC-7, pp. 1281–1295, Dec. 1989.

Improved Binary Codes and Sequence Families from Z_4-Linear Codes *

Abhijit G. Shanbhag
Communication Sciences Institute
University of Southern California
Los Angeles, CA 90089-2565, U.S.A.

P. Vijay Kumar
Communication Sciences Institute
University of Southern California
Los Angeles, CA 90089-2565, U.S.A.

Tor Helleseth
Department of Informatics
University of Bergen
Høyteknologisenteret, N-5020 Bergen, Norway

Abstract

A recently derived bound on exponential sums over Galois rings is used to construct a nested chain of Z_4-linear binary codes and binary sequences. When compared with the chain of Delsarte-Goethals' codes, the codes in the new chain offer a larger minimum distance for the same code size. A family of non-linear binary sequences having desirable correlation properties is then presented. For a given value of maximum correlation, the binary sequences have family size considerably larger than the best sequence families known.

I. Introduction

The Delsarte-Goethals codes $DG(m,r)$ [2], m odd, $m \geq 3$, $1 \leq r \leq \frac{m-1}{2}$, are a nested chain of non-linear binary codes which may be regarded as generalizations of the Kerdock code. It was shown recently (see [3]) that the Kerdock and Delsarte-Goethals codes are images under the Gray map of a chain $DG_4(m,r)$ of Z_4-linear codes.

The present paper uses a bound for exponential sums over Galois rings derived in [4] to construct a new nested chain of non-linear binary codes and sequences. The chain of codes are the Gray-mapped images of a new chain of Z_4-linear codes. The codes $DG(m,r)$ for $r = 1,2$ are also contained in the new code chain. Corresponding to each

*The work was supported in part by the National Science Foundation under Grant Number NCR-9016077, and in part by the Norwegian Research Council for Science and the Humanities.

code $DG(m,r)$, $3 \leq r \leq \frac{m-1}{2}$, we show that there is a code in the new chain, of the same length and size as $DG(m,r)$, but whose minimum distance is significantly larger.

By shortening the codes in the new chain of binary codes and then applying *Nechaev's* permutation one obtains equivalent *nonlinear cyclic* codes. These cyclic codes lead directly to a chain of efficient and nested sequence families. The first three members in the chain improve upon the small and large set of Kasami sequences and the modified Gold family constructed by Rothaus [7]. The first member in the chain of sequence families was independently constructed by Udaya and Siddiqi [8] who were unaware of the results of Nechaev. The succeeding members in the sequence chain are new.

Section II provides the background on Galois rings needed to discuss Z_4-linear codes. The new chain of non-linear binary codes is constructed and compared with the Delsarte-Goethals chain in Section III. The construction of the non-linear binary sequence families and comparisons with existing sequence families appears in Section IV.

II. Galois Rings of Characteristic 4

The Galois ring $R = GR(4,m)$ of size 4^m is a Galois extension of degree m over Z_4. R is a local ring having unique maximal ideal $M = 2R$. Let $\mu : R \to R/M$ denote the (mod 2) reduction map given by:

$$\mu(x) = x + M, \quad x \in R.$$

Thus, $\mu(R) = R/M \cong F$ where F is the finite field with 2^m elements.

As a multiplicative group, the units R^* in R have the following group structure:

$$R^* \cong Z_{2^m-1} \times \underbrace{Z_2 \times Z_2 \times \cdots \times Z_2}_{\text{m copies}}.$$

Let β be a *primitive* element of R^*, i.e. β is a generator of the cyclic group Z_{2^m-1}. Define $\mathcal{T} = \{0, 1, \beta, \cdots, \beta^{2^m-2}\}$. It can be shown that every element $z \in R$ has a unique 2-adic expansion

$$z = a + 2b, \quad a, b \in \mathcal{T}.$$

The Galois group of R/Z_4 is cyclic of order m generated by the automorphism σ given by

$$\sigma(x) = a^2 + 2b^2$$

where $x = a + 2b \in R$ and $a, b \in \mathcal{T}$. The trace mapping $Tr(\cdot) : R \to Z_4$ is then defined via

$$Tr(x) = x + \sigma(x) + \cdots + \sigma^{m-1}(x).$$

Let $f(x) \in R[x]$ be a *canonical* polynomial. By this we mean that

$$f(x) = \sum_{i=0}^{n} f_i x^i, \text{ with } f_i = 0 \text{ for } i \text{ even.}$$

We introduce the following degree-related parameters:

$$\begin{aligned} D_{1,f} &= \max\{i | 0 \le i \le n, f_i \in R \backslash 2R\}, \\ D_{2,f} &= \max\{i | 0 \le i \le n, f_i \in 2R, f_i \neq 0\}. \end{aligned}$$

Following [4] the *weighted degree* D_f of the polynomial $f(x)$ is defined via

$$D_f = \max\{2D_{1,f},\ D_{2,f}\}. \tag{1}$$

Note that when $f(x)$ is canonical, both $D_{1,f}$ and $D_{2,f}$ are odd. Theorem 1 in [4] then states:

Theorem 1.

$$\left| \sum_{x \in \mathcal{T}} \omega^{Tr(f(x))} \right| \le (D_f - 1)2^{\frac{m}{2}}, \quad \omega = \sqrt{-1}. \tag{2}$$

It is evident from Theorem 1 that

$$Re(\sum_{x \in \mathcal{T}} \omega^{Tr(f(x))}) \le (D_f - 1)2^{\frac{m}{2}}$$

where $Re(x)$ denotes the real part of x. However as shown below, it is possible to improve upon this bound in some specific instances.

Theorem 2. *Let $m \ge 3$ be odd. For a given integer r, $1 \le r \le \frac{m-1}{2}$, let $f(x) \in R[x]$, $f(x) \not\equiv 0$ be given as under*

$$f(x) = f_1 x + 2(\sum_{i=1}^{r} f_{1+2^i} x^{1+2^i}).$$

Then

$$Re(\sum_{x \in \mathcal{T}} \omega^{Tr(f(x))}) \le (D_f - 1)2^{\frac{m-1}{2}}.$$

Given an integer $D \geq 1$, let us define

$$S_D = \{f \in R[x] \mid D_f \leq D,\ f \text{ is canonical}\}.$$

Then we have:

Lemma 1. *Let $D \geq 1$ be an integer. Then*

$$|S_D| = 2^{(D-\lfloor \frac{D}{4} \rfloor)m}$$

where $\lfloor x \rfloor$ denotes the largest integer $\leq x$.

III. Codes Better than the Delsarte-Goethals Codes

A linear (n, M) quaternary code C_4 is a submodule of Z_4^n of size M. Let $\mathbf{a} = (a_0, \cdots, a_{n-1})$, $\mathbf{b} = (b_0, \cdots, b_{n-1})$ be any two codevectors in C_4. We define an inner-product on C_4 via

$$\rho(\mathbf{a}, \mathbf{b}) = \sum_{i=0}^{n-1} \omega^{a_i - b_i}, \quad \omega = \sqrt{-1}.$$

The *Lee* weights of 0,1,2,3 $\in Z_4$ are 0,1,2,1 respectively. The Lee norm of any vector in Z_4^n is the sum of the Lee weights of the components. We note that the Lee weight $w_L(y)$ of an element $y \in Z_4$ can be expressed as

$$w_L(y) = 1 - Re(\omega^y)$$

Thus, the Lee distance $d_L(\mathbf{a}, \mathbf{b}) = w_L(\mathbf{a} - \mathbf{b})$, $\mathbf{a}, \mathbf{b} \in C_4$, is related to $\rho(\mathbf{a}, \mathbf{b})$ via

$$d_L(\mathbf{a}, \mathbf{b}) = n - Re\{\rho(\mathbf{a}, \mathbf{b})\}. \tag{3}$$

Next, consider the (non-linear) map $\phi: Z_4 \to Z_2^2$:

$$0 \to 00,\ 1 \to 01,\ 2 \to 11,\ 3 \to 10.$$

This map, called the Gray map, can be viewed as the composite of two binary maps π, ν so that

$$\phi(x) = (\pi(x), \nu(x)), \quad x \in Z_4.$$

Extending ϕ in a natural way, we obtain a map

$$\begin{aligned} \phi: Z_4^n &\to Z_2^{2n} \\ \mathbf{c} &\to (\pi(\mathbf{c}), \nu(\mathbf{c})). \end{aligned}$$

Clearly ϕ is an isometry from Z_4^n (under the Lee metric) to Z_2^{2n} (under the Hamming metric).

Given a linear quaternary code C_4, let ρ_{max} denote maximum inner product between distinct codewords in C_4, i.e.

$$\rho_{max} = \max\{|\rho(\mathbf{a},\mathbf{b})| \mid \mathbf{a},\mathbf{b} \in C_4,\ \mathbf{a} \neq \mathbf{b}\}.$$

As a result of (3) and the isometry of the Gray map, we have:

Theorem 3. *Let C_4 be a Z_4-linear code and $C = \phi(C_4)$ be its image under the Gray map. Then the minimum distance d_{min} of C is given by:*

$$d_{min} = n - max\{Re(\rho(\mathbf{a},\mathbf{b})) \mid \mathbf{a},\mathbf{b} \in C_4,\ \mathbf{a} \neq \mathbf{b}\}.$$

Thus,

$$d_{min} \geq n - \rho_{max}$$

The Delsarte-Goethals codes $DG(m,r)$ [2], are a nested chain of non-linear binary codes which may be regarded as generalizations of the Kerdock code. It was shown recently (see [3]) that the Kerdock and the Delsarte-Goethals codes are the images under the Gray map of a chain $DG_4(m,r)$ of Z_4-linear codes. The following theorem provides the details.

Theorem 4. *Let $m \geq 3$ be odd and $1 \leq r \leq \frac{m-1}{2}$. Let $DG_4(m,r)$ denote the quaternary code*

$$DG_4(m,r) = \{\mathbf{c}_{\boldsymbol{\lambda},\epsilon} \mid \boldsymbol{\lambda} \in R \times \mathcal{T}^r,\ \epsilon \in Z_4\}$$

where the $(\boldsymbol{\lambda},\epsilon)^{th}$ codeword $\mathbf{c}_{\boldsymbol{\lambda},\epsilon} = (c_{\boldsymbol{\lambda},\epsilon}(x),\ x \in \mathcal{T})$ has the trace description

$$c_{\boldsymbol{\lambda},\epsilon}(x) = Tr(\lambda_0 x) + 2\sum_{i=1}^{r} Tr(\lambda_i x^{1+2^i}) + \epsilon,\quad \lambda_0 \in R,\ \lambda_i \in \mathcal{T},\ 1 \leq i \leq r.$$

The r^{th} element $DG(m,r)$, in the chain of binary, non-linear Delsarte-Goethals codes is then given by

$$DG(m,r) = \phi(DG_4(m,r))$$

and has parameters

$$length = 2^{m+1},\quad size = 4^{m+1}2^{rm},\quad d_{min} = 2^m - 2^{r-\frac{1}{2}} \cdot \sqrt{2^m}.$$

Using Theorem 1, we now construct a different nested chain of non-linear codes which are the Gray-map images of a new chain of Z_4-linear codes.

Given an integer D, $1 \leq D \leq 2^m - 1$, let $C_4(m, D)$ denote the Z_4-linear code:

$$C_4(m, D) = \{(Tr(f(x)) + \epsilon,\ x \in \mathcal{T}) \mid \epsilon \in Z_4,\ f(x) \in R[x], f \text{ canonical}, D_f \leq D\}.$$

The following Theorem follows from Lemma 1, Theorems 1 and 3:

Theorem 5. *Let $m \geq 3$ be odd. Let D be an integer satisfying $1 \leq D \leq 2^{\frac{m}{2}}$. Then the binary codes*

$$C(m, D) = \phi(C_4(m, D))$$

of length 2^{m+1} have parameters

$$d_{min} \geq 2^m - (D-1)2^{\frac{m}{2}}, \quad |C(m, D)| = 4 \cdot 2^{(D - \lfloor \frac{D}{4} \rfloor)m}.$$

It follows from Theorems 4 and 5 that for given D, the codes $DG(m, r)$ and $C(m, D)$ have the same size whenever

$$r = D - \lfloor \frac{D}{4} \rfloor - 2$$

or equivalently

$$D = \lfloor \frac{r+1}{3} \rfloor + r + 2.$$

Corollary 1. *Let $m \geq 3$ be odd. For a given integer r, $1 \leq r \leq \frac{m-1}{2}$, let the integer D_r be given by $D_r = \lfloor \frac{r+1}{3} \rfloor + r + 2$. Then the binary codes $DG(m, r)$ and $C(m, D_r)$ have the same size $4^{m+1}2^{rm}$, but different minimum distances d_{min} given by $2^m - 2^{r-\frac{1}{2}}2^{\frac{m}{2}}$ and $2^m - (\lfloor \frac{r+1}{3} \rfloor + r + 1)2^{\frac{m}{2}}$ respectively.*

Table 1 compares the minimum distance of the codes $DG(m, r)$ and $C(m, D_r)$ having the same size for $r = 3, 4, 5$ and 6. For values of $D_r = 2, 3, 5$, the codes $C(m, D_r)$ coincide with the Kerdock code and the first two members of the DG-chain respectively. However, we obtain a considerably higher (lower bound for) minimum distance for the succeeding members of $C(m, D_r)$ as compared to $DG(m, r)$ having the same size.

Table 1: A comparison of $C(m, D_r)$ with $DG(m, r)$ for same size of code ($m \geq 13$). In general, for any given value of r we require $m \geq 2r + 1$.

r	D_r	Common Code Size	$d_{min} \geq (C(m, D_r))$	d_{min} $(DG(m, r))$	Polynomial description of $C(m, D_r)$
3	6	2^{5m+2}	$2^m - (5)2^{\frac{m}{2}}$	$2^m - (4\sqrt{2})2^{\frac{m}{2}}$	$\epsilon + Ax + Bx^3 + 2Cx^5$
4	7	2^{6m+2}	$2^m - (6)2^{\frac{m}{2}}$	$2^m - (8\sqrt{2})2^{\frac{m}{2}}$	$\epsilon + Ax + Bx^3 + 2Cx^5 + 2Ex^7$
5	9	2^{7m+2}	$2^m - (8)2^{\frac{m}{2}}$	$2^m - (16\sqrt{2})2^{\frac{m}{2}}$	$\epsilon + Ax + Bx^3 + 2Cx^5 + 2Ex^7 + 2Fx^9$
6	10	2^{8m+2}	$2^m - (9)2^{\frac{m}{2}}$	$2^m - (32\sqrt{2})2^{\frac{m}{2}}$	$\epsilon + Ax + Bx^3 + Cx^5 + 2Ex^7 + 2Fx^9$

IV. New Family of Binary Sequences via Nechaev's Permutation

Henceforth we consider the codes $C_4^*(m, D)$ obtained by shortening $C_4(m, D)$ w.r.t. the coordinate $x = 0$. The binary code $C^*(m, D) = \phi(C_4^*(m, D))$ then has parameters:

$$\text{length} = 2^{m+1} - 2, \quad \text{size} = 2^{(D - \lfloor \frac{D}{4} \rfloor)m}, \quad d_{min} \geq 2^m - (D-1)2^{\frac{m}{2}}.$$

We now extend Nechaev's [6] analysis of the Kerdock code to show that the code $C^*(m, D)$ is equivalent via a permutation of coordinates, to a *cyclic* code.

Consider the binary code

$$P(m, D) = \{(\ \pi(3^t Tr[f(\beta^t)]),\ \ t = 0, 1, \cdots, 2^{m+1} - 3) \mid f \text{ canonical},\ D_f \leq D\}.$$

Let $Tr[f(\beta^t)] = a(t) + 2b(t)$, $a(t), b(t) \in \{0, 1\}$, $0 \leq t \leq 2^m - 2$, be the base-2 expansion of $Tr[f(\beta^t)]$. Since

$$\pi(3^t Tr[f(\beta^t)]) = \begin{cases} b(t) & t \text{ even} \\ b(t) \oplus a(t) & t \text{ odd} \end{cases} \tag{4}$$

where $0 \leq t \leq 2^{m+1} - 3$, it follows that the cyclic code $P(m, D)$ is equivalent to $C^*(m, D)$ under the permutation map η acting on the set $\{0, 1, \cdots, 2n - 1\}$, $n = 2^m - 1$, given by

$$(1, n+1)(3, n+3) \cdots (n-2, 2n-2).$$

The permutation map η will be referred to henceforth as Nechaev's permutation. The cyclic code $P(m, 2)$ is the punctured Kerdock code and was previously constructed by Nechaev [6].

We next extract from the cyclic code $P(m, D)$ a large family of binary sequences of period $2(2^m - 1)$ having desirable correlation properties. In the following lemma on periodicity of sequences, we denote the period of any sequence $y(t)$ by N_y.

Lemma 2. *Let $s(t)$ be a Z_4 sequence with period N_s, N_s odd, such that $\mu(s(t)) \not\equiv 0$. Define*

$$u(t) = 3^t s(t), \quad t \in Z$$

and

$$p(t) = \pi(u(t)), \quad t \in Z.$$

Then $u(t)$ and $p(t)$ have period $2 \cdot N_s$.

The following result follows from Lemma 2:

Corollary 2. *Let $f(x) = \sum_i f_i x^i \in R[x] \backslash 2R[x]$, $f(\cdot)$ canonical with $D_f \leq D$ for some integer $D \geq 1$. Set*

$$p(t) = \pi(3^t Tr[f(\beta^t)]), \quad \forall t \in Z.$$

Then $p(t)$ has period $2(2^m - 1)$ iff

$$g.c.d.\{\{i\}, 2^m - 1 \mid f_i \neq 0\} = 1.$$

Let the set

$$S(m, D) \subset P(m, D) \tag{5}$$

consist of a maximal family of pairwise, cyclically distinct sequences in $P(m, D)$ with each sequence having period $2(2^m - 1)$.

The principal result of this section is :

Theorem 6. *Given an odd integer $m \geq 3$ and an integer D satisfying $2 \leq D \leq 2^{\frac{m}{2}}$, let $S(m, D)$ be as defined in (5). The size $|S(m, D)|$ and maximum non-trivial correlation parameter $\zeta_{max}(m, D)$ of this family satisfy*

$$|S(m, D)| \geq 2^{(D - \lfloor \frac{D}{4} \rfloor - 1)m}, \quad \zeta_{max}(m, D) \leq 2 \cdot max\{Re\{\rho(\mathbf{a}, \mathbf{b})\} \mid \mathbf{a}, \mathbf{b} \in C_4^*(m, D)\} \tag{6}$$

$$\leq 2(D-1)2^{\frac{m}{2}} + 2. \tag{7}$$

Table 2 compares the family size and the correlation properties of $S(m, D)$ with some other sequences. Note that $S(m, 2)$, $S(m, 3)$ and $S(m, 5)$ are the images under the composition of the Gray map and Nechaev's permutation of Family $\mathcal{A}$ [1], $DG_4^*(m, 1)$ and $DG_4^*(m, 2)$ respectively. Theorem 2 and (6) are used to bound the maximum correlation of $S(m, 3)$ and $S(m, 5)$. As the correlation distribution of $S(m, 2)$ is known (see [1]), we again apply (6) to bound the maximum correlation of $S(m, 2)$. From Table 2 we

Table 2: Comparison of some families of sequences.

Family	Period L	Size of Family	Maximum Correlation Value	Normalized Maximum Correlation
$S(m, D)$	$2(2^m - 1)$	$\geq 2^{(D-\lfloor \frac{D}{4} \rfloor - 1)m}$ $\approx (\frac{L}{2})^{(D-\lfloor \frac{D}{4} \rfloor - 1)}$	$\leq 2(D-1)2^{\frac{m}{2}} + 2$	$\leq (D-1)(2L)^{\frac{1}{2}}$
Kasami (Small Set)	$2^m - 1$, m even	$2^{\frac{m}{2}} \approx L^{\frac{1}{2}}$	$2^{\frac{m}{2}} + 1$	$\approx L^{\frac{1}{2}}$
$S(m, 2)$	$2(2^m - 1)$, m odd	$2^m \approx \frac{L}{2}$	$\sqrt{2} \cdot 2^{\frac{m}{2}} + 2$	
Kasami (Large Set)	$2^m - 1$, m even	$2^{\frac{m}{2}}(2^m + 1) \approx L^{\frac{3}{2}}$	$2 \cdot 2^{\frac{m}{2}} + 1$	$\approx 2 \cdot L^{\frac{1}{2}}$
$S(m, 3)$	$2(2^m - 1)$, m odd	$2^{2m} \approx (\frac{L}{2})^2$	$2\sqrt{2} \cdot 2^{\frac{m}{2}} + 2$	
Modified Gold (Rothaus)	$2^m - 1$, m even	$2^{\frac{m}{2}} \frac{2^{3m}-1}{2^m-1} \approx L^{\frac{5}{2}}$	$4 \cdot 2^{\frac{m}{2}} + 1$	$\approx 4 \cdot L^{\frac{1}{2}}$
$S(m, 5)$	$2(2^m - 1)$,m odd	$2^{3m} \approx (\frac{L}{2})^3$	$4\sqrt{2} \cdot 2^{\frac{m}{2}} + 2$	

note that $S(m, 2)$, $S(m, 3)$ and $S(m, 5)$ have the same maximum correlation (normalized by length) but a larger family size by a factor of $\sqrt{L}$ as compared to the Small and Large sets of Kasami sequences and the modified Gold sequences [7] respectively. It is straightforward to construct Family $S(m, 2)$ from Nechaev's construction of a cyclic code equivalent to the punctured Kerdock code. Udaya and Siddiqi [8] who were unaware of the results of Nechaev present an independent construction of Family $S(m, 2)$.

References

[1] S. Boztas, A.R. Hammons, Jr., and P.V. Kumar. "4-Phase Sequences with Near-Optimum Correlation Properties". *IEEE Trans. Inform. Theory*, 38(3):1101–1113, May 1992.

[2] P. Delsarte and J. M. Goethals. "Alternating Bilinear Forms over GF(q)". *Journal Combinatorial Theory*, 19:26–50, 1975.

[3] A.R. Hammons, P.V. Kumar, A.R. Calderbank, N.J.A. Sloane, and P. Solé. "The Z_4-Linearity of Kerdock, Preparata, Goethals and Related Codes". *IEEE Trans. Inform. Theory*, March 1994.

[4] P.V. Kumar, T. Helleseth, and A.R. Calderbank. "An Upper Bound for some Exponential Sums over Galois Rings and Applications". to appear in the *IEEE Trans. Inform. Theory*, 1994.

[5] F.J. MacWilliams and N.J.A. Sloane. *"The Theory of Error-Correcting Codes"*. North-Holland, Amsterdam, 1977.

[6] A.A. Nechaev. "The Kerdock Code in a Cyclic Form". *Discrete Math.*, 1:123–139, 1989.

[7] O. Rothaus. "Modified Gold Codes ". *IEEE Trans. Inform. Theory*, 39:654–656, March 1993.

[8] P. Udaya and M.U. Siddiqi. "Optimal and Suboptimal Biphase Sequences with Controllable Complexity Derived from the Proper Ideal of Z_4 ". Preprint, 1994.

On the Cardinality of the Group of Permutations that Commute with the Discrete Fourier Transform*

Rüdiger Urbanke, Quinn Li and Bixio Rimoldi†
ruediger@vienna.wustl.edu, quinn@shannon.wustl.edu, bixio@essrl.wustl.edu
Washington University, Department of Electrical Engineering
Electronics Systems and Signals Research Laboratory
St. Louis, MO 63130, USA

Abstract

Let F be a field (finite or infinite) such that the Discrete Fourier Transform of length n is defined. The exact number C_n of permutations σ that commute with the Discrete Fourier Transform is derived. As application to coding theory, such permutations associate to every cyclic code a set of equivalent cyclic codes.

Index terms: Discrete Fourier Transform, Cyclic Codes, Permutations

I Introduction

Let F be a field (finite or infinite). Assume that F contains an nth root of unity so that the Discrete Fourier Transform (DFT) of length n is defined (see [1]). In [2] the conditions for a permutation σ to commute with the DFT over the complex field were stated. In [3] it was shown that if $(n, m) = 1$, where (n, m) denotes the greatest common divisor of n and m, then $C_{nm} = C_n C_m$, where C_n denotes the number of permutations that commute with the DFT. In this paper, we derive the exact number of such permutations.

*An extended and more detailed version of this paper is available by the authors.

†Supported by National Science Foundation grant NCR-9109944, NCR-9357689.

It is well known that a (n,k) cyclic code over a finite field F can be described as the set of codewords $x \in \mathsf{F}^n$ that have specified $n-k$ zeros in the Fourier domain. Hence, a permutation $\sigma : \{0,1,\cdots,n-1\} \to \{0,1,\cdots,n-1\}$ that commutes with the DFT sends a cyclic code to an equivalent cyclic code since these two codes differ only by a permutation of the coordinates (in the time as well as in the frequency domain). This is one reason for our interest in permutations that commute with the DFT.

In the sequel, N denotes the non-negative integers.

II Main Part

The main result is summarized in the following

Theorem 1 For $n \in \mathsf{N}$, $n \geq 2$, the number C_n of permutations that commute with the Discrete Fourier Transform of length n is given by

$$C_n = 2^{\psi(n)+\eta(\nu_2)}\,, \tag{1}$$

where $\psi(n)$ is the number of distinct factors in the prime factorization of $n = \prod p^{\nu_p}$ and

$$\eta(m) = \begin{cases} 0 & m = 0 \\ -1 & m = 1 \\ 0 & m = 2 \\ 1 & m \geq 3\,. \end{cases} \tag{2}$$

Example 1 For $n = 9 = 3 * 3$, we have $\psi(9) = 1$ and $\nu_2 = 0$ and hence $C_9 = 2$. For $n = 24 = 2^3 * 3$, we have $\psi(24) = 2$ and $\nu_2 = 3$ and hence $C_{24} = 2^3 = 8$.

In the sequel we will give an outline of the proof. Let σ be a permutation on n letters $0,\cdots,n-1$, and σ^{-1} be its inverse. As was shown in [2] and [3] for $\mathsf{F} = \mathsf{C}$ the following are two equivalent (necessary and sufficient) conditions for a permutation σ to commute with the DFT:

(1) $\sigma(i)\sigma(k) = ik \bmod n$ for all $0 \leq i,k \leq n-1$.

(2) $[\sigma(1)]^2 = 1 \bmod n$ and $\sigma(k) = k\sigma(1) \bmod n$ for all $0 \leq i,k \leq n-1$.

It is straightforward to extend the proof to the case of finite fields with the only additional requirement that a DFT of length n must be defined, i.e., there must exists an element $\alpha \in \mathsf{F}$ which is an n-th root of unity.

If we let $x = \sigma(1)$ and interpret x as element of the ring $\mathsf{Z}/n\mathsf{Z}$, then we see from condition (2) that $x^2 = e$, where e denotes the unit element of the ring. Hence x must

be an element of order 1 or 2. On the other hand one can also show that each element of $\mathbb{Z}/n\mathbb{Z}$ of order 1 or 2 defines exactly one permutation that commutes with the DFT. This is summarized in

Lemma 1 The number of permutations that commute with the DFT equals the number of elements x in the ring $\mathbb{Z}/n\mathbb{Z}$ such that $x^2 = e$.

Observe that if $x \in \mathbb{Z}/n\mathbb{Z}$ satisfies $x^2 = e$, then x has a multiplicative inverse. Hence, we can restrict our search for solutions to $x^2 = e$ in $\mathbb{Z}/n\mathbb{Z}$ to all *units*, i.e., invertible elements, of $\mathbb{Z}/n\mathbb{Z}$. This is the set $(\mathbb{Z}/n\mathbb{Z})^*$, which can be characterized as the set of $x \in \mathbb{Z}/n\mathbb{Z}$ with x relatively prime to n. We can verify that $(\mathbb{Z}/n\mathbb{Z})^*$ is an abelian multiplicative group (see [4, p.95]) with $\phi(n)$ elements, where $\phi(n)$ is the Euler phi function (see [4, p. 71]).

Lemma 2 Let G be a group and assume that G can be written as the direct product $G = G_1 \times G_2$. Then the number of elements in G of order 1 or 2 equals the product of the number of such elements in G_1 and G_2.

Proof. Let $g \in G$. Then by assumption $g = (g_1, g_2)$ for some $g_1 \in G_1$ and $g_2 \in G_2$. Hence

$$g^2 = e^2 \Leftrightarrow (g_1^2, g_2^2) = (e_1, e_2)$$

which holds if and only if $g_1^2 = e_1$ and $g_2^2 = e_2$. □

Lemma 2 together with the fact that (see [4, p. 71])

$$(\mathbb{Z}/n\mathbb{Z})^* \approx \prod_p (\mathbb{Z}/p^{\nu_p}\mathbb{Z})^*$$

shows that C_n is a multiplicative function of n, i.e., $C_{nm} = C_n C_m$ if $(n, m) = 1$. Hence, it suffices to study the special cases C_{p^j} for $j \in \mathbb{N}$.

The key of the proof lies now in the fact that if G is a cyclic group with odd $|G|$ then it has a single element of order 1 (and no element of order 2) and if $|G|$ is even then it has exactly one element of order 1 and one element of order 2. The first fact follows by observing that the order of an element must divide the cardinality of the group, the second fact can be deduced from looking at the homomorphism $\pi : G \to G$ given by $\pi(g) = g^2$.

These facts can be used as follows. For the cases $p \geq 3$ and $j \geq 1$ we have (see [4, p.72])

$$(\mathbb{Z}/p^j\mathbb{Z})^* \approx (\mathbb{Z}/p\mathbb{Z})^* \times U_1$$

where $U_1 = 1 + p\mathbb{Z} \bmod p^j$. Further, U_1 is a cyclic group of order p^{j-1} (which is odd). Note that $(\mathbb{Z}/p\mathbb{Z})$ is a finite field and, hence, $(\mathbb{Z}/p\mathbb{Z})^*$ is a cyclic group of order $p - 1$ (which is even). It follows that $(\mathbb{Z}/p\mathbb{Z})^*$ has exactly 2 elements of order 1 or 2 and that

U_1 has 1 such element. Hence, by Lemma 2, $(\mathsf{Z}/p^j\mathsf{Z})^*$ has exactly two elements such that $x^2 = e$.

The case C_{2^j} can be solved in a similar way by observing that for $j \geq 2$ (see [4, p.72])

$$(\mathsf{Z}/2^j\mathsf{Z})^* \approx \{1, -1\} \times U_2$$

where $U_2 = 1 + 4\mathsf{Z} \bmod 2^j$ is a cyclic group of order 2^{j-2}. We get

$$C_{2^j} = \begin{cases} 1 & j = 1 \\ 2 & j = 2 \\ 4 & j \geq 3 \,. \end{cases}$$

This, together with the multiplicativity of C_n proofs Theorem 1.

Cyclic codes can be described as the set of n-length codewords over GF(q) such that their DFT has zeros at specific positions (see [1]). As a permutation does not change the Hamming weight of a codeword, we see that each permutation that commutes with the DFT defines a (not necessarily distinct) cyclic code having exactly the same weight distribution, and whose zero positions in the Fourier domain are permutations of those of the original code.

Example 2 Let F = GF(2), E = GF(2^3). Then a DFT of length $n = 7$ is defined. A single-error correcting binary code of length 7 is then given by choosing $\{1, 2, 4\}$ to be the set of zeros in the Fourier domain. This is, in fact, the well-known $(7, 4, 3)$ binary Hamming code. For $n = 7$ there exists exactly one non-trivial permutation that commutes with the DFT which is given by $\sigma(1) = 6$. We see that the set of zeros of the original code is carried into the set $\{3, 5, 6\}$. Note that $\{1, 2, 4\}$ and $\{3, 5, 6\}$ are both conjugacy classes, and that σ carries one conjugacy class into the other. We will now show that this is always the case.

Lemma 3 A permutation that commutes with the DFT carries conjugacy classes into conjugacy classes. More precisely, the conjugacy class $\{ip^k \bmod n :\in \mathsf{N}\}$ will be mapped into the conjugacy class $\{\sigma(i)p^k \bmod n :\in \mathsf{N}\}$.

Proof. As stated, the elements of a conjugacy class have the form $ip^k \bmod n$, $k \in \mathsf{N}$, where p is the characteristic of the field E. But if $0 \leq i < n - 1$, then $\sigma(ip^k \bmod n) = (i\sigma(1)p^k \bmod n) = (\sigma(i)p^k \bmod n)$ by condition (2). The claim follows. □

References

[1] R. Blahut, *Theory and Practice of Error Control Codes*, Addison-Wesley, Reading, Massachusetts, 1983.

[2] C. Y. Chao, "On a type of circulants," *Linear Algebra and its Applications*, vol. 6, pp. 241–248, 1973.

[3] P. J. S. G. Ferreira, "A group of permutations that commute with the Discrete Fourier Transform," *IEEE Transactions on Signal Processing*, vol. 42, pp. 444–445, February 1994.

[4] S. Lang, *Undergraduate Algebra*, Springer Verlag, New York, 1990.

OPTIMALITY AND INTERCHANGEABILITY OF LEAKY BUCKETS

TAKIS KONSTANTOPOULOS *
DEPT. OF ECE
UNIVERSITY OF TEXAS
AUSTIN, TX 78712
takis@alea.ece.utexas.edu

VENKAT ANANTHARAM †
DEPT. OF EECS
UNIVERSITY OF CALIFORNIA
BERKELEY, CA 94720
ananth@eecs.berkeley.edu

Abstract

Some aspects of recent work of the authors centering around the leaky bucket are summarized. The problem of regulating offered traffic to an ATM network is considered. The flow controller should be burst reducing in that the admitted traffic satisfies prescribed burstiness constraints. Both single and multiple burstiness constraints are considered. For a single burstiness constraint, the optimal flow controller is the leaky bucket. For multiple burstiness constraints a series arrangement of leaky buckets is optimal and an equivalent fork/join parallel implementation is proposed. In particular leaky buckets in series are seen to be interchangeable. A novel use of the concept of reflection mapping is central to the optimality proofs.

1 Introduction

In this paper we summarize recent work of the authors on the optimal design of flow control schemes for offered traffic to an integrated services network [1], [6], focusing on those results that relate to the leaky bucket scheme. Such networks are expected to use

*Research supported in part by NSF grant NCR 92-11343
†Research supported in part by NSF grant NCR 88-57731

Asynchronous Transfer Mode (ATM) so we refer to them also as ATM networks. Offered traffic to an ATM network is expected to include traffic of a highly bursty nature that may not be well modeled by the traditionally employed traffic models (Poisson, renewal, Markov modulated processes, etc.). We thus study the flow control problem with a much wider class of models for offered traffic that does not make assumptions about its statistics.

Traffic processes are modeled as sigma-finite measures on $\mathbf{R}_+$. No stationarity or other such stochastic assumptions are made. Our models may thus be thought of as fluid models, although they allow for jumps. Before offered traffic is admitted to the network it needs to be regulated to meet burstiness constraints. For the notion of burstiness we use the definition of Cruz [3, 4]. A traffic process A is said to be (σ, ρ) constrained if

$$A[s,t] \leq \sigma + \rho(t-s) \ ,$$

for all $t \geq s \geq 0$, where $A(s,t]$ denotes the amount of traffic over the interval $(s,t]$. The pair of parameters σ, ρ form a simple and appealing characterization of burstiness that capture important aspects of the proposed ATM standards.

Our focus is the design of flow controllers that ensure that traffic admitted to the network obeys burstiness constraints in the above sense. The first question is how a user, constrained to offer a flow obeying (σ, ρ) constraints, can optimally control his source to ensure this requirement. The design problem in general will involve a number of other constraints; both *delay* or *buffer* constraints are considered in [6]. A delay constraint places an upper bound on the maximum delay that a cell can tolerate between its arrival and its transmission or refusal by the flow controller. A buffer constraint places a bound on the total buffer capacity that can be used by the flow controller.

A second set of problems involves multiple burstiness constraints. A traffic process A is simultaneously (σ_i, ρ_i) constrained, $i = 1, \ldots, n$, if

$$A[s,t] \leq \min_{1 \leq i \leq n} (\sigma_i + \rho_i(t-s)) \ ,$$

for all $t \geq s \geq 0$. By varying the parameters σ_i, ρ_i, the network manager has more degrees of freedom and flexibility in restricting the nature of admitted flow to the network.

In all these problem formulations we have determined flow control designs that are optimal, in the sense that the volume of admitted traffic is as large as possible subject to the constraints, [1], [6]. It is very satisfying that the optimal flow control is very simple, involving recursively updating a quantity which we call *virtual backlog*. Thus each proposed optimal design is easy to implement: recursive formulas are readily available. What is more, for the problem with a single burstiness constraint and no additional buffer or delay constraint, the popular leaky bucket flow control scheme is seen to be optimal. For multiple burstiness constraints, tandems of appropriately dimensioned leaky buckets in series are seen to be optimal, and can be conveniently realized with an equivalent fork-join implementation. ¿From a mathematical point of view the optimality proofs involve a novel use of the so-called reflection mapping using time-varying boundaries. For full details for all these statments, the reader is referred to [1], [6], the first of which is currently available as a preprint, and the second as a technical report.

Here we wish to give a few more details about those of our results that relate to the leaky bucket scheme. The paper is organized as follows. In Section 2 we discuss

the optimal flow controller that meets a single burstiness constraint subject to a delay constraint. In Section 3 this is related to the leaky bucket by relaxing the delay constraint. In Section 4 we discuss the optimal flow controller for multiple burstiness constraints. This is related to tandems of leaky buckets and the alternate fork-join implementation is discussed. We also comment on the extension to the problem with general concave increasing constraints. A quick review of reflection mappings and their properties is given in the Appendix.

2 Flow control with delay constraints

Let $\mathcal{M}$ be the set of all sigma-finite measures on $\mathbf{R}_+$. Sigma-finiteness excludes the possibility of an infinite volume of traffic on a bounded interval. Denote by A^S the restriction of $A \in \mathcal{M}$ on the set $S \subseteq \mathbf{R}_+$ and let $A_t = A[0,t]$. Note that $t \to A_t$ is increasing and right continuous and with left limits. Also, $A(s,t] = A_t - A_s$. The space $\mathcal{M}$ includes, as special cases, point processes (processes describing cells arriving at discrete time epochs), as well as absolutely continuous processes (processes with well defined rate).

Consider the problem of designing a flow controller that regulates an offered traffic stream so that its output (the admitted traffic stream) satisfies a (σ, ρ) burstiness constraint and such that offered traffic is delayed by no more than d before being admitted (delay constraint). To this end, we define a *feasible flow controller* to be a map $\varphi : \mathcal{M} \to \mathcal{M}$ satisfying the following requirements.

I. It is realizable: $B_t \leq A_t$, for all $t \geq 0$, $A \in \mathcal{M}$, $B = \varphi(A)$ (i.e. it can only transmit past flow).

II. It is causal: $\varphi(A_1^{[0,t]} + A_2^{(t,\infty)})^{[0,t]} = \varphi(A_1)^{[0,t]}$, for all $t \geq 0$, $A_1, A_2 \in \mathcal{M}$ (i.e. its decisions are based only on the past of the arrival process).

III. It is (σ, ρ) constrained with initial virtual backlog σ_0: $B(s,t] \leq \sigma + \rho(t-s)$, $B_t \leq \sigma_0 + \rho t$, for all $t \geq s \geq 0$, $A \in \mathcal{M}$, $B = \varphi(A)$.

IV. It poses a bound d on the delay that an incoming traffic unit can tolerate: $B(s,t] \leq A(s-d,t]$, for all $t \geq s \geq 0$, $A \in \mathcal{M}$, $B = \varphi(A)$.

Note that every $A \in \mathcal{M}$ is taken to be zero on $(-\infty, 0)$.

A feasible flow controller is said to be *optimal* if for any other feasible flow controller $\bar{\varphi}$ and any traffic process $A \in \mathcal{M}$,

$$\varphi(A)_t \geq \bar{\varphi}(A)_t \,,$$

for all $t \geq 0$.

In [6], we show that an optimal feasible flow controller exists and give an explicit construction. The optimal flow controller can be understood in terms of reflection mappings. The reader is referred to the Appendix for a quick review of their definition and basic properties.

The optimal flow controller is described as follows. Let

$$\psi_t = A(t-d,t], \beta_t = \sigma + \psi_t \ . \tag{1}$$

For any arrival process $A \in \mathcal{M}$ consider the reflection of the "free process"

$$x_t = \sigma - \sigma_0 + A_t - \rho t \tag{2}$$

in the time-varying interval $[0, \beta_t]$. Denote by $q = \mathcal{R}(x), \ell = \mathcal{L}(x), u = \mathcal{U}(x)$ the reflected, upper and lower processes respectively. Thus

$$q_t = x_t + \ell_t - u_t \ . \tag{3}$$

Finally define the map $\varphi : \mathcal{M} \to \mathcal{M}$ by $B = \varphi(A)$, where

$$B_t = A_t - u_t - (q_t - \sigma)^+ \ . \tag{4}$$

¿From (2) and (3) we see that B is also given by

$$B_t = \sigma_0 + \rho t - \ell_t - (\sigma - q_t)^+ \tag{5}$$

and also by

$$B_t = \begin{cases} A_t - u_t & \text{if } q_t \leq \sigma \\ \sigma_0 + \rho t - \ell_t & \text{otherwise.} \end{cases} \tag{6}$$

The following theorem is proved in [6].

Theorem 1 *The map $\varphi : \mathcal{M} \to \mathcal{M}$ defined via (1)-(6) above is feasible and optimal.*

The most important feature of the flow controller proposed above is that it is easily implementable in real time. For instance, if arrivals occur in discrete time, with, say, $A_n = a_0 + \cdots + a_{n-1}$, $n = 0, 1, 2, \ldots$, then the amount of traffic b_n transmitted on the n-th time slot is given by the following algorithm:

$$\begin{aligned} b_n &= (a_n + \lambda_n) \wedge (r_n + \sigma_n) \\ \lambda_{n+1} &= (\lambda_n + a_n - b_n) \wedge (a_{n-d+1} + \cdots + a_n) \\ \sigma_{n+1} &= \sigma \wedge (\sigma_n + r_n - b_n) \end{aligned}$$

The quantity σ_n is the virtual backlog at time n and λ_n is the amount of traffic stored in the flow controller's memory at time n.

3 Leaky bucket flow control

Let us now drop the delay constraint. Thus we now say that a flow controller is feasible if it satisfies requirements I, II and III of the previous section. The optimal flow controller is now described by the following procedure: Reflect the free process x upwards at zero (the upper boundary now is at infinity since $d = \infty$) and let $q = \mathcal{R}(x)$, $\ell = \mathcal{L}(x)$. Then

$$q_t = \sigma - \sigma_0 + A_t - \rho t + \ell_t \ . \tag{7}$$

The optimal flow controller is the map $\varphi : \mathcal{M} \to \mathcal{M}$, $B = \varphi(A)$, where

$$B_t = A_t - (q_t - \sigma)^+ . \tag{8}$$

We see that B is also given by

$$B_t = \sigma_0 + \rho t - \ell_t - (\sigma - q_t)^+ \tag{9}$$

We now describe the so-called leaky bucket flow controller. A (σ, ρ) leaky bucket operates as follows. Admission of offered traffic is controlled by a stream of *tokens*. The token stream is a process of constant rate ρ, and arrives into a *token buffer* that has size σ. Assume that initially the token buffer contains an amount of token equal to σ_0. Offered traffic that finds tokens in the token buffer is transmitted instantaneously: one can visualize the traffic picking up an equal amount of token and being admitted into the network. Offered traffic that does not find tokens in the token buffer is stored in a *cell buffer* of infinite capacity and released as tokens arrive. Tokens arriving when the token buffer is full are lost. The leaky bucket is a popular and effective scheme to regulate the burstiness of source traffic before it is admitted to an ATM network.

To see that the map φ constructed by (7)-(9) above is indeed the input-output map of the (σ, ρ) leaky bucket one may argue as follows. From the verbal description of the (σ, ρ) leaky bucket we see that cells and tokens cannot simultaneously be present in the system. Thus the state of the system can be encapsulated by the single quantity "$\sigma +$ cell buffer size $-$ token buffer size ". Further, tokens are lost by virtue of the token buffer being full precisely when this quantity is 0. We may now see that q_t in the mathematical description corresponds precisely to this quantity. Indeed q_t can be written as $q_t = \sigma + (q_t - \sigma)^+ - (\sigma - q_t)^+$. The process ℓ_t corresponds to the increasing process of lost tokens, because it increases "at rate ρ" precisely when $q_t = 0$, from the description of the reflection mapping. ¿From (8) we see that $(q_t - \sigma)^+ = A_t - B_t$. Since A_t is the amount of traffic that arrived before t, while B_t is the amount of traffic transmitted by time t, the quantity $(q_t - \sigma)^+$ is the amount of traffic stored in the flow controller's cell buffer at time t. Similarly, (9) implies $(\sigma - q_t)^+ = \sigma_0 + (\rho t - \ell_t) - B_t$. The quantity $(\sigma - q_t)^+$ can then be interpreted as amount of token stored in the token buffer at time t. Indeed, the last expression is simply a token-balance equation: σ_0 is the initial amount of token; the amount of token departed by time t is precisely equal to the number B_t of cells departed by time t; the amount of token arrived by time t is equal to ρt minus the amount of token ℓ_t lost due to the overflow of the token buffer. Note that at any point of time t the amount of traffic in the cell buffer $(q_t - \sigma)^+$ and the amount of token in the token buffer $(\sigma - q_t)^+$ cannot be simultaneously positive. Further, one can also see from the decomposition property that $\rho t - \ell_t$, which is the process of useful token, is a increasing process as it should be. This comes directly from applying the decomposition property to the expression $x_t = \sigma - \sigma_0 + A_t - \rho t$, which expresses the free process as a difference of increasing processes.

4 General burstiness constraints

We now turn into the problem of identifying an optimal flow controller that, besides being realizable and causal (see Section 2), produces an output traffic process $B = \varphi(A)$

such that

$$B_t - B_s \leq f(t-s)\ , B_t \leq f_0(t)\ ,$$

where $f_0, f : \mathbf{R}_+ \to \mathbf{R}_+$ are concave increasing functions, and $f_0 \leq f$.

We first examine the case where the functions f_0, f are both piecewise linear:

$$f(t) = \min_{1 \leq i \leq n} (\sigma_i + \rho_i t)\ ,$$
$$f_0(t) = \min_{1 \leq i \leq n} (\sigma_{i0} + \rho_i t)\ ,$$

where $\sigma_{i0} \leq \sigma_i$, and $\rho_i \geq 0$ for all i.

In [1] the following was shown to be true:

Theorem 2 *Let φ^i be a (σ_i, ρ_i) leaky bucket with initial token amount σ_{i0}, $i = 1, \ldots, n$. Let $\varphi = \varphi^n \circ \varphi^{n-1} \circ \cdots \circ \varphi^1$. Then*

- *The map φ is an optimal flow controller.*
- *φ is the composition of φ^i, $i = 1, \ldots, n$ in any order.*

In other words, the overall output process from n leaky buckets placed in series is the same regardless of their order. Furthermore, this series arrangement is an optimal flow controller whose output satisfies multiple burstiness constraints.

¿From an implementation point of view the following result is useful. See [1] for a proof.

Theorem 3 *Let φ^i be a leaky bucket as above. Then, for any $A \in \mathcal{M}$,*

$$\varphi^n \circ \cdots \circ \varphi^1(A) = \varphi^1(A) \wedge \cdots \wedge \varphi^n(A)\ .$$

This theorem says that the tandem configuration of n leaky buckets is equivalent to a parallel one. Namely, let n leaky buckets be fed by the same process A and let their outputs be directed into a "join". One can describe this algorithm as follows (suppose that $n = 2$ for simplicity). Let the tokens in leaky bucket 1 be red and tokens in leaky bucket 2 blue. When a cell arrives into the flow controller it is split into two copies which are simultaneously sent to each leaky bucket. Thus the first copy will pick up a red token if it is available and depart instantaneously, or will wait in the cell buffer. The first copy eventually departs from the first leaky bucket by carrying a red token. Similarly, the second copy eventually departs from the second leaky bucket by carrying a blue token. Suppose that the copy carrying a red cell departs first from its leaky bucket. Then it is directed towards another buffer where it waits till the first moment of time that the copy carrying a blue cell departs from its leaky bucket. At that moment the two copies depart simultaneously, and physically the flow controller triggers the transmission of a cell into the network. The final stage of the system is a "join", performing the synchronization operation just described. The join provides a buffer for storing copies of cells carrying colored tokens. It can be seen that at all points of time at least one of the buffers of the join is empty.

Finally, the solution of the flow control problem for arbitrary functions f_0, f is given by the following procedure.

Let f_0^*, f^* be the convex duals of f_0, f, respectively. The convex dual of f is defined by

$$f^*(\rho) = \sup_{t \geq 0}[f(t) - \rho t] .$$

Since $f_0 \leq f$, we have $f_0^* \leq f^*$. The optimal flow controller maps $A \in \mathcal{M}$ into $B \in \mathcal{M}$ defined by the following equations:

$$x_t^\rho := f^*(\rho) - f_0^*(\rho) + A_t - \rho t \tag{10}$$

$$\ell^\rho := \mathcal{L}(x^\rho) \tag{11}$$

$$B_t := A_t \wedge \inf_{\rho \geq 0}[f_0^*(\rho) + \rho t - \ell_t^\rho] . \tag{12}$$

Observe that, for each ρ,

$$B_t^\rho := A_t \wedge [f_0^*(\rho) + \rho t - \ell_t^\rho]$$

is the output of a $(\sigma = f^*(\rho), \rho)$ leaky bucket with initial token amount $\sigma_0 = f_0^*(\rho)$. Also, for any finite set $J = \{\rho_1, \ldots, \rho_n\}$, the process

$$B_t^J := A_t \wedge \min_{\rho \in J}[f_0^*(\rho) + \rho t - \ell_t^\rho]$$

is the output of a tandem configuration of $(\sigma_i = f^*(\rho_i), \rho_i)$ leaky buckets with initial token amounts $\sigma_{0i} = f_0^*(\rho_i)$, $i = 1, \ldots, n$ (regardless of the order). The flow controller defined by equations (10)-(12) above is feasible and optimal.

Letting $Q_t = A_t - B_t$ be the total number of cells stored in the flow controller's buffers at time t (where B is given by expression (12) above), we find the following formula:

$$Q_t = \sup_{0 \leq s \leq t}[(A_t - A_s) - f(t-s)] \vee [A_t - f_0(t)] \vee 0 .$$

This has a finite dimensional causal realization if the functions f_0, f are piecewise linear with finitely many pieces.

5 Appendix

Reflection mappings are used in order to define the flow controllers as well as in establishing their optimality. We review the definition of the reflection mapping of a "free process" x_t in a–possibly time-varying–interval $I_t = [\alpha_t, \beta_t]$. We denote by $D_A[0, \infty)$ the collection of all càdlàg (i.e., right-continuous with left limits) functions mapping $[0, \infty)$ into a space A, a subset of $\mathbf{R}$ for our purposes. One of the earliest references to reflection in one dimension, in a stochastic context, is due to Skorokhod [7]. See also Harrison [5] for a nice discussion of the reflection with two boundaries and for continuous x, Williams [2], and the references therein.

The reflection problem

We are given càdlàg functions x_t (the free process), α_t (the lower boundary), and β_t (the upper boundary). Denote by I_t the interval $[\alpha_t, \beta_t]$. (Usually, I_t is constant.) It is assumed that $\alpha_t < \beta_t$ for all $t \geq 0$ (strict inequality is important if additional conditions are not to be imposed), and $\alpha_0 \leq x_0 \leq \beta_0$. The case where one of the boundaries is at infinity is allowed. The–so called–Skorokhod's reflection problem looks for functions ℓ_t (*lower boundary process*), u_t (*upper boundary process*) such that

1. The *reflected process*
$$q_t := x_t + \ell_t - u_t$$
is in I_t for all t.

2. Both ℓ_t, u_t are nondecreasing with $\ell_0 = u_0 = 0$ and ℓ_t (u_t) increases only when $q_t = \alpha_t$ ($q_t = \beta_t$), i.e.,
$$\int_{\mathbf{R}_+} 1(q_t > \alpha_t) d\ell_t = 0 \ , \quad \int_{\mathbf{R}_+} 1(q_t < \beta_t) du_t = 0.$$

It turns out that the solution to this problem is unique and can be constructed by successive approximations (c.f. Harrison [5]). The mapping $x \mapsto (\ell, u, q)$ is referred to as "the reflection mapping". Our notation is

$$q = \mathcal{R}(x), \quad \ell = \mathcal{L}(x), \quad u = \mathcal{U}(x) \ .$$

Formulas for single boundary reflection

There are no explicit formulas when both boundaries are present. However, when only one boundary is present (i.e., when either $\alpha_t = -\infty$ or $\beta_t = +\infty$) we do have useful explicit formulas, which are worthwhile writing down.

Lower boundary only

$$\begin{aligned} \ell_t &= -\inf_{0<s\leq t} (x_s - \alpha_s) \wedge 0 \ , \\ q_t = x_t + \ell_t &= \sup_{0<s\leq t} (x_t - x_s + \alpha_s) \vee x_t \ . \end{aligned}$$

Upper boundary only

$$\begin{aligned} u_t &= \sup_{0<s\leq t} (x_s - \beta_s) \vee 0 \ , \\ q_t &= \inf_{0<s\leq t} (x_t - x_s + \beta_s) \wedge x_t \ . \end{aligned}$$

Fundamental properties

The following properties of the reflection mapping are used in the optimality proofs of the various flow controllers (see [1, 6]).

1. **Causality.** Fix time s and consider the free process after s, defined by

$$x_t^s := q_s + x_{s+t} - x_s, \ t \geq 0.$$

To obtain the reflected process at t time units after s, we may forget entirely the past of x, q before s and consider the reflection of the new free process $\{x_t^s, \ t \geq 0\}$, i.e.,

$$q_{s+t} = x_t^s + \mathcal{L}(x^s)_t - \mathcal{U}(x^s)_t \ .$$

2. **Minimality.** The boundary process ϑ_t that maintains q_t within the constraint set is the smallest process (in the sense of smallest variation) that manages to do so. In other words, if ϑ_t' is another process that is associated with q_t with respect to the same constraint set and direction field, then $|\vartheta|_t \leq |\vartheta|_t'$, for all t. To put it in terms of the ℓ and u, suppose that ℓ_t', u_t' are two increasing processes such that

$$q_t' := x_t + \ell_t' - u_t' \in [\alpha_t, \beta_t], \text{ for all } t \geq 0.$$

Then

$$\ell_t \leq \ell_t', \ \ u_t \leq u_t', \text{ for all } t \geq 0.$$

3. **Monotonicity.** When two free processes x, x' compare in the sense that

$$x_t - x_s \leq x_t' - x_s', \ \ 0 \leq s \leq t \ ,$$

then the reflected process compare in the sense that

$$q_t \leq q_t', \ \ t \geq 0 \ .$$

4. **Decomposition.** Suppose that x_t is of bounded variation itself. Then x_t can be written as the difference of two increasing processes:

$$x_t = x_0 + A_t - B_t \ ,$$

where A_t, B_t are increasing processes, starting from zero. Reflect x_t in the interval $[\alpha_t, \beta_t]$:

$$q_t = x_0 + (A_t - u_t) - (B_t - \ell_t) \ .$$

Then $A_t - u_t$, $B_t - \ell_t$ are increasing processes. In this case we may think of the reflection mapping as a mapping from a pair of (positive) measures into another pair of (positive) measures, i.e., $(A, B) \mapsto (A - u, B - \ell)$. This point of view is useful.

References

[1] Anantharam, V. and Konstantopoulos, T. (1994). Optimality and interchangeability of leaky buckets in tandem. Preprint.

[2] Chung, K.L. and Williams, R.J. (1983). *Introduction to Stochastic Integration.* Birkhäuser, Boston.

[3] Cruz, R.L. (1991). A calculus for network delay, part I: network elements in isolation. *IEEE Tr. Inf. Th.*, **37**, 114-131.

[4] Cruz, R.L. (1991). A calculus for network delay, part II: network analysis. *IEEE Tr. Inf. Th.*, **37**, 132-141.

[5] Harrison, J.M. (1985). *Brownian Motion and Stochastic Flow Systems.* Wiley, New York.

[6] Konstantopoulos, T., and Anantharam, V. (1994). Optimal Flow Control Schemes that Regulate the Burstiness of Traffic. *SCC Technical Report SCC-93-15*, ECE Department, University of Texas, Austin (submitted to *IEEE Transactions on Networking*).

[7] Skorokhod, A.V. (1961). Stochastic equations for diffusion processes with boundaries. *Th. Prob. Appl.*, **6**, 287-298.

Architectures for Lightweight Buffered Packet Switches *

R. L. Cruz J. T. Tsai
Department of Electrical & Computer Engineering
University of California, San Diego
La Jolla, CA 92093-0407
cruz@ece.ucsd.edu om@cts4.ucsd.edu

Abstract

In an earlier paper, we introduced "COD" architectures for high speed optical packet multiplexing. In this paper we utilize similar ideas in order to synthesize buffered packet switching architectures, which have both multiplexing and demultiplexing functionality. The architectures were conceived for use with opto-electronic switches and fiber optic delay lines, but may also suitable for "lightweight" VLSI implementations of buffered packet switches. We first consider packet switches with two input ports and two output ports. We then use these as building blocks to construct buffered multichannel switches (i.e. "Buffered Statistical Data Forks").

1 Building Blocks

We consider a discrete time model for packet switching, where all packets are of equal length. Time is divided into equal length slots with duration equal to the transmission time of a single packet.

1.1 The TC Module

We will utilize a basic building block, called the "Track-Changer" (TC) module, which is illustrated in Figure 1. The TC module has two input ports and two output ports, and there is no buffering of packets. Packets entering on the input ports can be of two "types," type 0 and type 1. Type 0 packets "wish" to be routed to the bottom output (marked with a 0), and type 1 packets wish to be routed to the top output (marked with a 1). If only one packet arrives to the TC module in a slot, the TC module routes the packet to its desired output. If two packets arrive during a slot, the TC module attempts to route the packets to their desired outputs. If the two packets are of different types, no conflict occurs and the packets leave their desired outputs. If both packets are of type

*This work was supported by the National Science Foundation under the grants NCR 93-18416 and NCR 91-58618.

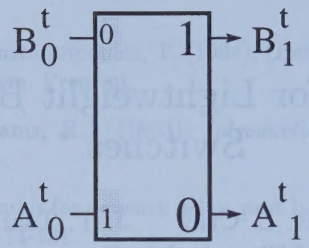


Figure 1: The Track Changer (TC) module.

0, the packet entering the top input (marked with a 0) receives priority and leaves the bottom output, and the other packet is "deflected" to the top output. Similarly, if both packets are of type 1, the packet entering the bottom input (marked with a 1) receives priority and leaves the top output, and the other packet is "deflected" to the bottom output.

The name "Track Changer" comes from an analogy with trains traveling on two tracks. Trains of type 1 want the top track and trains of type 0 want the bottom track. In case of a conflict, the train which wants to change tracks receives priority.

The TC module can be thought of as a 2×2 packet switch with no buffering. The control of a TC module is slightly more complicated than the "LC" module introduced in [2] (for the construction of multiplexers), where all packets are of type 0. In the case of the LC module, control signals for routing can be determined simply by detecting the presence of packets. The TC module, however, must read "header" information in a packet to determine the type of the packet. The ports of a TC module could carry packets in an optical or electronic format.

We will use "indicator" random variables to describe the status of ports in each slot. If a variable is equal to 0,1, or ϕ, that corresponds to a packet of type 0, type 1, or no packet at all traveling on the corresponding port during the corresponding slot, respectively. For example, in Figure 1 we use the variables A_0^t, B_0^t, A_1^t, and B_1^t, to describe the bottom input, top input, bottom output, and top output of a TC module during slot t, respectively. We then have, with slight abuse of notation,

$$\begin{aligned}
\{A_1^t = 0\} &= \{A_0^t = 0 \cup B_0^t = 0\} \\
\{A_1^t = 1\} &= \{A_0^t = 1 \cap B_0^t = 1\} \\
\{B_1^t = 0\} &= \{A_0^t = 0 \cap B_0^t = 0\} \\
\{B_1^t = 1\} &= \{A_0^t = 1 \cup B_0^t = 1\} .
\end{aligned}$$

The following is trivial, but useful to note for future reference:

$$\{B_1^t = 0\} \subset \{A_1^t = 0\} \quad (1)$$

$$\{A_1^t = 1\} \subset \{B_1^t = 1\} . \quad (2)$$

In words, this means a packet of type 0 leaving the top output implies that a packet of type 0 leaves the bottom output during the same slot. Similarly, a packet of type 1 leaving the bottom output implies that a packet of type 1 leaves the top output during the same slot.

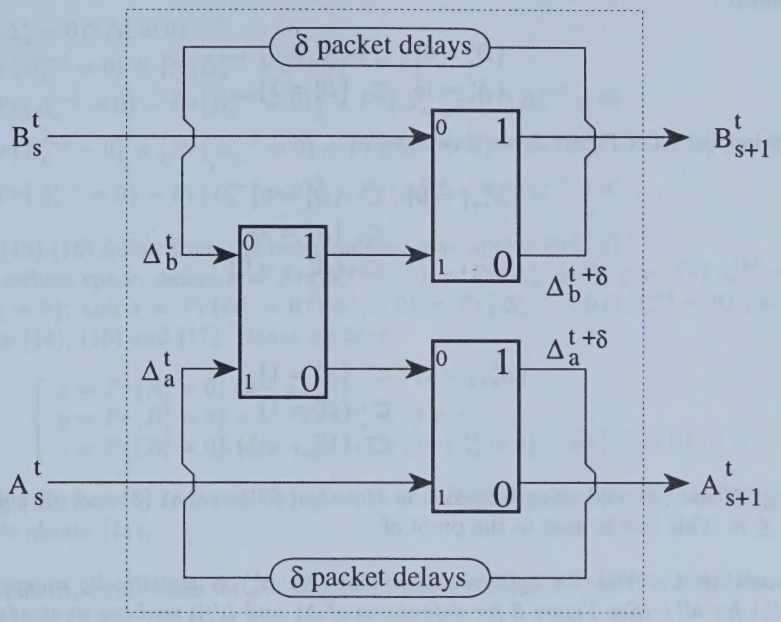


Figure 2: The Joined Twin - Recursive Track Changer (JT-RTC) module.

1.2 The JT-RTC Module

Next, we introduce the "Joined Twin - Recursive Track Changer" (JT-RTC) module, which is illustrated in Figure 2.

As the figure indicates, the JT-RTC module uses the TC module (3 of them) as a building block, as well as two delay lines in a feedback (recursive) configuration. Each delay line holds packets for δ slots, where δ is an integer. Note that the JT-RTC module "attempts" to route type 0 packets towards the bottom output and type 1 packets towards the top output, and can be thought of as a buffered packet switch.

Apparently, a smaller rate of deflections (packet leaving on un-desired outputs) can result if the delay lines in a given stage are of different lengths. However, we set the lengths equal here since that makes the performance analysis more tractable.

2 A Lightweight Buffered 2×2 Packet Switch

As the notation in Figures 1 and 2 suggests, consider the series interconnection of a TC module (on the left), followed by n JT-RTC modules. The notation in Figure 2 is for the s^{th} JT-RTC module in the chain, and we allow the length δ of each of the delay lines in stage s to vary with s. If the delay line lengths are chosen carefully, we will see that the resulting cascade connection of JT-RLC modules can result in a remarkably efficient buffered packet switch.

Suppose that the input processes to the TC module on the left, $\{A_0^t\}$ and $\{B_0^t\}$, are

ergodic random processes. In order to begin the analysis, suppose that (see Figure 2 for notation)

$$\{B_s^t = 0\} \subset \{A_s^t = 0\} \tag{3}$$
$$\{A_s^t = 1\} \subset \{B_s^t = 1\} . \tag{4}$$

By inspection of Figure 2, we thus have

$$\{B_{s+1}^t = 0\} \subset \{B_s^t = 0\} \tag{5}$$
$$\subset \{A_s^t = 0\} \tag{6}$$
$$\subset \{A_{s+1}^t = 0\} \tag{7}$$

and

$$\{A_{s+1}^t = 1\} \subset \{A_s^t = 1\} \tag{8}$$
$$\subset \{B_s^t = 1\} \tag{9}$$
$$\subset \{B_{s+1}^t = 1\} . \tag{10}$$

Noting (1) and (2), and using recursion in s, it then follows that (3) and (4) hold for all $1 \leq s \leq n$. This fact is used in the proof of

Proposition 1 *With the approximation that (Δ_b^t, Δ_a^t) is statistically independent of (B_s^t, A_s^t) for all t (See Figure 2 for definitions of Δ_b^t and Δ_a^t), we have in steady state*

$$Pr\{B_{s+1}^t = 0\} = \frac{(Pr\{B_s^t = 0\})^3}{(1 - Pr\{A_s^t = 0\})^2 + Pr\{B_s^t = 0\}(1 + Pr\{B_s^t = 0\} - Pr\{A_s^t = 0\})} \tag{11}$$

and by conservation of flow, $Pr\{A_{s+1}^t = 0\} = Pr\{A_s^t = 0\} + Pr\{B_s^t = 0\} - Pr\{B_{s+1}^t = 0\}$.

Proof: By inspection of Figure 2 again, we have

$$\begin{aligned}
\{B_{s+1}^t = 0\} &= \{B_s^t = 0\} \cap \{\Delta_a^t = 0\} \cap \{\Delta_b^t = 0\} \\
\{\Delta_a^{t+\delta} = 0\} &= \{A_s^t = 0\} \cap \left(\{\Delta_a^t = 0\} \cup \{\Delta_b^t = 0\}\right) \\
\{\Delta_b^{t+\delta} = 0\} &= \{B_s^t = 0\} \cup \left(\{\Delta_a^t = 0\} \cap \{\Delta_b^t = 0\}\right) \\
\{\Delta_a^t = 0\} \cap \{\Delta_b^t = 0\} &= \left[\{A_s^{t-\delta} = 0\} \cap \left(\{\Delta_a^{t-\delta} = 0\} \cup \{\Delta_b^{t-\delta} = 0\}\right)\right] \cap \\
&\quad \left[\{B_s^{t-\delta} = 0\} \cup \left(\{\Delta_a^{t-\delta} = 0\} \cap \{\Delta_b^{t-\delta} = 0\}\right)\right] .
\end{aligned} \tag{12}$$

Taking the probability measure of each of the above equations, we obtain

$$Pr\{B_{s+1}^t = 0\} = Pr\{B_s^t = 0\} \times Pr\{\Delta_a^t = 0 \cap \Delta_b^t = 0\} \tag{13}$$
$$Pr\{\Delta_a^{t+\delta} = 0\} = Pr\{A_s^t = 0\} \times \left(Pr\{\Delta_a^t = 0\} + Pr\{\Delta_b^t = 0\} - Pr\{\Delta_a^t = 0 \cap \Delta_b^t = 0\}\right) \tag{14}$$
$$Pr\{\Delta_b^{t+\delta} = 0\} = Pr\{B_s^t = 0\} + Pr\{\Delta_a^t = 0 \cap \Delta_b^t = 0\} - Pr\{B_s^t = 0\} \times Pr\{\Delta_a^t = 0 \cap \Delta_b^t = 0\} \tag{15}$$

and

$$
\begin{aligned}
& Pr\{\Delta_a^t = 0 \cap \Delta_b^t = 0\} \\
&= Pr\{B_s^{t-\delta} = 0\} \times Pr\{\Delta_a^{t-\delta} = 0 \cup \Delta_b^{t-\delta} = 0\} + \\
&\quad \Big(Pr\{A_s^{t-\delta} = 0\} - Pr\{B_s^{t-\delta} = 0\}\Big) \times Pr\{\Delta_a^{t-\delta} = 0 \cap \Delta_b^{t-\delta} = 0\} \qquad (16) \\
&= Pr\{B_s^{t-\delta} = 0\} \times \Big(Pr\{\Delta_a^{t-\delta} = 0\} + Pr\{\Delta_b^{t-\delta} = 0\} - Pr\{\Delta_a^{t-\delta} = 0 \cap \Delta_b^{t-\delta} = 0\}\Big) + \\
&\quad \Big(Pr\{A_s^{t-\delta} = 0\} - Pr\{B_s^{t-\delta} = 0\}\Big) \times Pr\{\Delta_a^{t-\delta} = 0 \cap \Delta_b^{t-\delta} = 0\} \ , \qquad (17)
\end{aligned}
$$

where (13)-(16) follow from the independence assumption and (3).

To reduce space, define $x = Pr\{\Delta_a^{t+\delta} = 0\} = Pr\{\Delta_a^t = 0\}$; $y = Pr\{\Delta_b^{t+\delta} = 0\} = Pr\{\Delta_b^t = 0\}$; and $z = Pr\{\Delta_a^t = 0 \cap \Delta_b^t = 0\} = Pr\{\Delta_a^{t-\delta} = 0 \cap \Delta_b^{t-\delta} = 0\}$ and apply these to (14), (15) and (17). Hence we have

$$
\begin{cases}
x = Pr\{A_s^t = 0\}\,(x + y - z) \\
y = Pr\{B_s^t = 0\} + z - Pr\{B_s^t = 0\} \times z \\
z = Pr\{B_s^t = 0\} \times (x + y - z) + (Pr\{A_s^t = 0\} - Pr\{B_s^t = 0\}) \times z
\end{cases} \qquad (18)
$$

Solving (18) for z in terms of $Pr\{A_s^t = 0\}$ and $Pr\{B_s^t = 0\}$ and then applying this to (13), we obtain (11).

$\diamondsuit$

Symmetric expressions for $Pr\{A_{s+1}^t = 1\}$ and $Pr\{B_{s+1}^t = 1\}$ can be obtained in terms of $Pr\{A_s^t = 1\}$ and $Pr\{B_s^t = 1\}$.

It is intuitively clear that the independence approximation in the hypothesis of Proposition 1 can be made arbitrarily accurate by choosing the lengths of the delay lines in stage s much larger than the "statistical memory length" of the arrival process to stage s, $\{B_s^t, A_s^t\}$. We also believe that the independence approximation might be made reasonably accurate by choosing the δ's in each stage to be relatively prime integers.

Define $\gamma = Pr\{B_1^t = 0\}/(1 - Pr\{A_1^t = 0\})$ and $\nu = Pr\{A_1^t = 1\}/(1 - Pr\{B_1^t = 1\})$. The recursion in s of Proposition 1 can be solved neatly in closed form:

Corollary 1 *If the hypothesis of Proposition 1 holds in each stage s, there holds*

$$
Pr\{B_{n+1}^t = 0\} = Pr\{B_1^t = 0\}(1 - \gamma)\frac{\gamma^{(3^n - 1)}}{1 - \gamma^{(3^n)}} \qquad (19)
$$

$$
1 - Pr\{A_{n+1}^t = 0\} = (1 - Pr\{A_1^t = 0\})(1 - \gamma)\frac{1}{1 - \gamma^{(3^n)}} \qquad (20)
$$

and similarly

$$
Pr\{A_{n+1}^t = 1\} = Pr\{A_1^t = 1\}(1 - \nu)\frac{\nu^{(3^n - 1)}}{1 - \nu^{(3^n)}} \qquad (21)
$$

$$
1 - Pr\{B_{n+1}^t = 1\} = (1 - Pr\{B_1^t = 1\})(1 - \nu)\frac{1}{1 - \nu^{(3^n)}} \ . \qquad (22)
$$

Proof: Define $\gamma_s = \dfrac{Pr\{B_s^t = 0\}}{1 - Pr\{A_s^t = 0\}}$, and note that $\gamma = \gamma_1$. Equation (11) in Proposition 1 becomes

$$
Pr\{B_{s+1}^t = 0\} = Pr\{B_s^t = 0\} \times \frac{\gamma_s^2}{1 + \gamma_s + \gamma_s^2} \ . \qquad (23)
$$

Since $Pr\{A_{s+1}^t = 0\} = Pr\{A_s^t = 0\} + Pr\{B_s^t = 0\} - Pr\{B_{s+1}^t = 0\}$, we have

$$\begin{aligned}
1 - Pr\{A_{s+1}^t = 0\} &= 1 - Pr\{A_s^t = 0\} - Pr\{B_s^t = 0\} + Pr\{B_{s+1}^t = 0\} \\
&= \Big(1 - Pr\{A_s^t = 0\}\Big)\left(1 - \gamma_s + \frac{Pr\{B_{s+1}^t = 0\}}{1 - Pr\{A_s^t = 0\}}\right) \\
&= \Big(1 - Pr\{A_s^t = 0\}\Big)\left(1 - \gamma_s + \frac{\gamma_s^3}{1 + \gamma_s + \gamma_s^2}\right) \qquad (24) \\
&= \Big(1 - Pr\{A_s^t = 0\}\Big) \times \frac{1}{1 + \gamma_s + \gamma_s^2} \qquad (25)
\end{aligned}$$

where (24) follows from (23). Dividing (23) by (25), we obtain

$$\gamma_{s+1} = (\gamma_s)^3 \quad \Longrightarrow \quad \gamma_{s+1} = \gamma^{3^s} . \qquad (26)$$

From (25), we have

$$\begin{aligned}
1 - Pr\{A_{s+1}^t = 0\} &= \Big(1 - Pr\{A_1^t = 0\}\Big) \times \prod_{i=1}^{s} \frac{1}{1 + \gamma_i + \gamma_i^2} \\
&= \Big(1 - Pr\{A_1^t = 0\}\Big) \frac{1 - \gamma}{(1 - \gamma_1)(1 + \gamma_1 + \gamma_1^2)} \times \prod_{i=2}^{s} \frac{1}{1 + \gamma_i + \gamma_i^2} \\
&= \Big(1 - Pr\{A_1^t = 0\}\Big) \frac{1 - \gamma}{(1 - \gamma_2)(1 + \gamma_2 + \gamma_2^2)} \times \prod_{i=3}^{s} \frac{1}{1 + \gamma_i + \gamma_i^2} \\
&\cdots\cdots\cdots \\
&= \Big(1 - Pr\{A_1^t = 0\}\Big) \frac{1 - \gamma}{(1 - \gamma_s)(1 + \gamma_s + \gamma_s^2)} \\
&= \Big(1 - Pr\{A_1^t = 0\}\Big) \frac{1 - \gamma}{1 - \gamma_{s+1}} . \qquad (27)
\end{aligned}$$

We have used (26) and $(1 - \gamma_i)(1 + \gamma_i + \gamma_i^2) = 1 - \gamma_i^3 = 1 - \gamma_{i+1}$ in the above to get (27). Multiplying (27) by γ_{s+1}, we obtain

$$Pr\{B_{s+1}^t = 0\} = \frac{Pr\{B_1^t = 0\}}{\gamma} \times \gamma_{s+1} \times \frac{1 - \gamma}{1 - \gamma_{s+1}} . \qquad (28)$$

Applying (26) to (28) and (27), we get (19) and (20). With symmetry, (21) and (22) follow similarly.

$\Diamond$

Note that $Pr\{B_{n+1}^t = 0\}$ is the rate at which type 0 packets leave the "wrong" (i.e. the top) output, and is the "loss rate" for type 0 packets. Assuming $\gamma < 1$, Corollary 1 implies that $Pr\{B_{n+1}^t = 0\} = O(\gamma^{(3^n)})$, which is striking. In fact this is the loss rate of a "standard" work conserving packet switch with a buffer capacity of $3^n - 1$ packets which is fed by memoryless traffic. This means that n need not be large in order to yield very small loss (deflection) probabilities.

It it interesting to note that the results above depend on the distribution of packet arrivals only through γ and ν. Intuitively, though, the more "bursty" the arrival process is, the longer the delay lines will have to be in order for the independence approximations to be accurate.

Note that a packet cannot travel through any delay line more than once, so that the delay of any packet is upper bounded by the sum of the lengths of all delay lines.

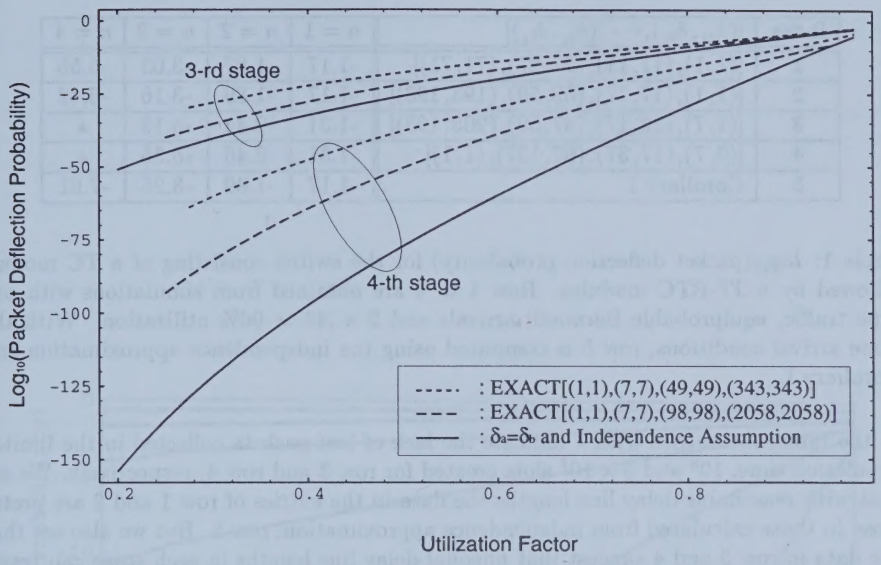


Figure 3: $\log_{10}$(packet deflection probability) versus utilization factor.

3 Numerical results and possible applications

Figure 3 plots $\log_{10}$(packet deflection probability) versus utilization factor with $n = 3$ and $n = 4$, where arrivals of each packet type are assumed to be equiprobable Bernoulli on each input link. The solid curves result from Corollary 1 which exploits the independence approximation. The two curves above each solid line are exact results, which are obtained from computations based on the underlying Markovian property. Given a buffered 2×2 switch constructed by the cascade of a TC module followed by n JT-RTC modules, where the s^{th} JT-RTC module has delay line lengths δ_{a_s} and δ_{b_s} for the bottom and top segments, respectively(See Figure 2), we say the delay line profile is $[(\delta_{a_1}, \delta_{b_1}), (\delta_{a_2}, \delta_{b_2}), \ldots, (\delta_{a_n}, \delta_{b_n})]$. In this figure, exact values are given for the delay line profiles of $[(1,1),(7,7),(7^2,7^2),(7^3,7^3)]$ and $[(1,1),(7,7),(2\times 7^2, 2\times 7^2),(6\times 7^3, 6\times 7^3)]$. These exact results were computed by solving a system of 87465 and 524790 linear equations, respectively. Details of the analysis are omitted. Note that to reduce the number of equations, we have deliberateley chosen the delay line length of one stage to be an integer multiple of that of any previous stage. As can be seen, increasing the lengths of the delay lines in the fourth stage from 343 to 2058 improves the accuracy of the independence approximation.

The numerical example above suggests that the delay lines must grow rather quickly to yield results consistent with the independence approximation. However, there are heuristics for choosing the delay line profile which may yield smaller delay lines for the same performance. One such heuristic is to choose the delay line lengths in different stages to be *relatively prime* numbers. This was applied to our simulations, the results of which are listed in row 1 to 4 of Table 1. In the simulations, packets of one type arriving at each input are equiprobable Bernoulli and the system has $2 \times .48 = 96\%$ utilization.

Index	$[(\delta_{a_1}, \delta_{b_1}), \cdots, (\delta_{a_4}, \delta_{b_4})]$	$n = 1$	$n = 2$	$n = 3$	$n = 4$
1	$[(1,1),(11,11),(27,27),(71,71)]$	-1.17	-1.87	-3.03	-5.55
2	$[(1,1),(17,17),(59,59),(193,193)]$	-1.17	-1.89	-3.16	-6.42
3	$[(1,7),(23,17),(47,59),(205,193)]$	-1.31	-2.44	-5.13	$\star$
4	$[(5,7),(17,37),(67,137),(1,1)]$	-1.31	-2.46	-5.35	$\star$
5	Corollary 1	-1.17	-1.89	-3.25	-7.01

Table 1: log_{10}(packet deflection probability) for the switch consisting of a TC module followed by n JT-RTC modules. Row 1 to 4 are obtained from simulations with one type traffic, equiprobable Bernoulli arrivals and $2 \times .48 = 96\%$ utilization. With the same arrival conditions, row 5 is computed using the independence approximation and Corollary 1.

In the table, the entries with $\star$ indicate the lack of lost packets collected in the limited simulation time, 10^8 and 5×10^7 slots created for row 3 and row 4, respectively. We see that with reasonable delay line lengths the data in the entries of row 1 and 2 are pretty close to those calculated from independence approximation, row 5. But we also see that the data in row 3 and 4 suggest that unequal delay line lengths in each stage can result in much better loss probability performance than with equal delay line lengths.

We can use the buffered 2×2 packet switches presented in the previous section as building blocks for larger packet switches. For example, we can use these buffered 2×2 switches to build multistage "banyan" switching fabrics.

In particular, we can construct multichannel switches using the concept of "error correction" in the Statistical Data Fork [1]. In [3], buffers were introduced into the Statistical Data Fork. Using the approach presented in the previous section may yield an efficient way to implement a large buffering capability into each 2×2 switch.

As an example, Figure 4 plots the base 10 logarithm of the routing error probability versus link utilization factor for the case of a 64 port switch with 4 equally loaded output groups of size 16 each. The "codewords" used (see [1]) were 000000, 011011, 101101, and 110110, and the output ports were assigned optimally to groups at each utilization. There are six curves given in the figure, corresponding to $n = 0, 1, 2, 3, 4, 5$. Recall that n is the number of JT-RTC stages used in each 2×2 switching system. Thus, $n = 0$ corresponds to no buffering at all (the case discussed in [1]). It is seen, for example in the case of $n = 5$ JT-RTC modules used, that the routing error probability remains below 10^{-10} for all utilizations below 97%.

References

[1] R. L. Cruz, "The Statistical Data Fork: A Class of Broadband Multichannel Switches," *IEEE Trans. Commun.*, vol. 40 no. 10, Oct. 1992.

[2] R. L. Cruz and J. T. Tsai, "COD: Architectures for Optical Time-Based Multiplexers," *Proc. Allerton Conference*, Oct. 1993.

[3] B. R. Bellur and G. H. Sasaki, "A Buffered Statistical Data Fork," to appear in *Performance Evaluation.*

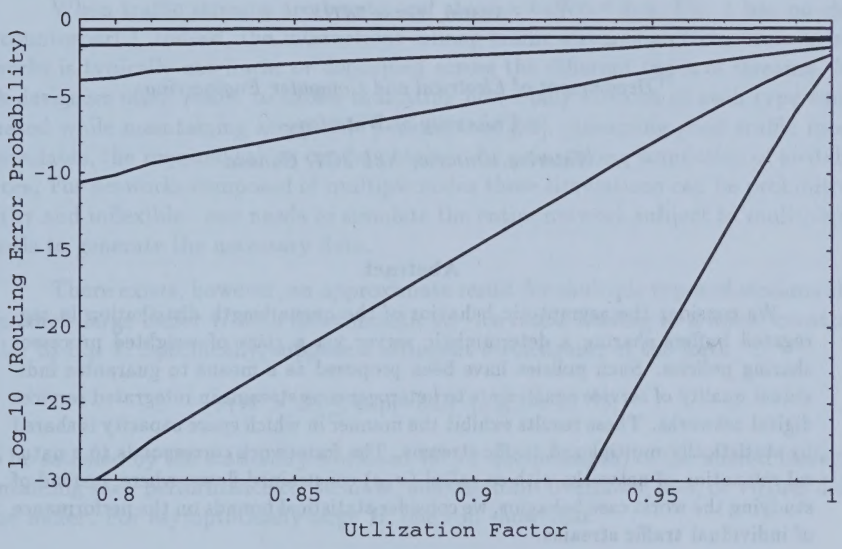


Figure 4: Performance of a 64 port Buffered Statistical Data Fork with 4 Equally Loaded Trunk Groups.

Bandwidth Allocation for Multiple Qualities of Service using Generalized Processor Sharing*

G. de Veciana† and G. Kesidis††

† *Department of Electrical and Computer Engineering*
University of Texas at Austin
Austin, Texas 78712

†† *Department of Electrical and Computer Engineering*
University of Waterloo
Waterloo, Ontario, N2L 3G1, Canada

Abstract

We consider the asymptotic behavior of the queue length distribution in segregated buffers sharing a deterministic server via a class of weighted processor sharing policies. Such policies have been proposed as a means to guarantee individual quality of service constraints to heterogeneous streams in integrated services digital networks. These results exhibit the manner in which spare capacity is shared by statistically multiplexed traffic streams. The framework corresponds to a natural relaxation of networks with so called (σ, ρ)-constrained flows, where, instead of studying the worst case behavior, we consider statistical bounds on the performance of individual traffic streams.

1 Introduction

ATM-based BISDN networks with heterogeneous applications requiring stringent performance guarantees will need appropriate service provisioning schemes including: buffer / bandwidth allocation, call admission, and call routing [10, 21]. The difficulty of this problem relative to traditional (circuit-switched) telephone networks lies in the multiplexing of multiple types of packetized traffic streams and messages via switches and communication links. In order for streams to share resources, one must guard against traffic fluctuations by inserting buffers. This in turn introduces interactions among relatively bursty traffic streams often rendering performance analysis difficult and the ensuing traffic management schemes inefficient.

*The research of G. de Veciana was supported in part by the National Science Foundation under Grant NCR-9409722. The research of G. Kesidis was supported by NSERC of Canada. A complete version of this paper is available by email request: *gustavo@mocha.ece.utexas.edu*

To ease the task of managing such a network it is desirable to obtain an equivalent circuit-switched model based on archetypes for different classes of streams; e.g., audio, graphics, low and high quality TV, and LAN-to-LAN traffic. For example, suppose a collection of sources, n_j of type $j \in J$, each known to require a *fixed* bandwidth α_j, share a link with capacity c. One can easily check for available spare bandwidth by considering whether

$$\sum_{j \in J} n_j \alpha_j \leq c. \tag{1}$$

Such a scheme, when extended to a network, would resemble traditional telephone systems where connections are set up if physical resources are available to link the source to the destination. Standard techniques used in telephony could then be adapted to packet-switched networks.

When traffic streams are bursty and share a *buffered* link, Eq. 1 has no clear-cut counterpart.[1] Indeed, the interactions among traffic streams and resources in such networks is typically not linear or decoupled across the different types of streams. Network designers often resort to tables indicating how many streams of each type can be tolerated while maintaining acceptable performance [20]. Assuming good traffic models are available, the required tables can be obtained by painstaking simulation of switching devices. For networks composed of multiple nodes these simulations can be prohibitively lengthy and inflexible - one needs to simulate the entire network subject to multiple sets of loads to generate the necessary data.

There exists, however, an approximate result for multiple types of streams sharing a *single* large buffer B with deterministic service rate c leading to a linear constraint similar to Eq. 1. Specifically, suppose a stringent δ-constraint of the form

$$\mathbb{P}(W > B) \leq \exp[-\delta B] \ \ (\text{e.g.}, \approx 10^{-9}), \tag{2}$$

is to be satisfied by the stationary workload W (or queue length) of the shared buffer; by maintaining such performance constraints, one can limit overflows, loss, or virtual delays in the buffer. For asymptotically large B, one can show that

$$\sum_{j \in J} n_j \alpha_j(\delta) \leq c \Leftrightarrow \lim_{B \to \infty} \frac{1}{B} \log \mathbb{P}(W > B) \leq -\delta, \tag{3}$$

where $\alpha_j(\delta)$, the *effective bandwidth* of stream j, depends on the statistics of the traffic stream and ranges from the mean to the peak rate of the traffic type as δ increases. If a stream has slotted arrivals $\{A_n^j\}$, its effective bandwidth is given by

$$\alpha_j(\delta) = \frac{\Lambda_j(\delta)}{\delta}, \ \ \Lambda_j(\delta) = \lim_{n \to \infty} \frac{1}{n} \log \mathbb{E} \exp[\delta \sum_{i=1}^{n} A_i^j], \tag{4}$$

and can be computed for most of the standard traffic models. The constraint on the right hand side of Eq. 3 does not strictly guarantee that Eq. 2 is satisfied; however, it does suggest that for large enough B this might be roughly the case. Assuming such results are valid, call acceptance can be easily carried out by checking if the available

[1] A conservative approach would be to let α_j denote the peak rate of stream j, while an optimistic one would instead assign mean rate. Neither of these approaches is truly useful when traffic is bursty.

capacity is larger than the effective bandwidth of the new stream. Numerous researchers have suggested the use of such results for bandwidth allocation. For early work in this area see [15, 13, 14]. The existence of effective bandwidths for Markov fluid sources was studied via spectral expansions in [12, 11]. General results were obtained via large deviations in [16, 6, 22].

Although this approach is appealing, many further issues need to be addressed before it becomes viable. Our goal herein is to investigate network designs oriented towards *individual* quality of service (QoS) guarantees rather than aggregate performance constraints. Notice that guaranteeing a cell loss probability no worse than 10^{-9} to the aggregate of a *heterogeneous* mix of traffic streams sharing a buffer is not equivalent to making the same performance guarantee to each of the streams individually [9]. If, however, statistically identical streams share a buffer then the aggregate and individual loss constraints are the same.

In order to guarantee *individual* users specific QoS requirements, buffer and bandwidth sharing rules partially isolating their streams from one another must be put into place. A typical approach is to have traffic queue in segregated buffers while sharing the output capacity via service policies roughly corresponding to weighted round robin; for example, dynamic-time slicing [20] and Generalized Processor Sharing (GPS) [18]. The GPS concept in conjunction with leaky bucket flow control has proven to be a particularly successful framework in which to guarantee individual worst case end-to-end backlog or delay bounds [19]. In this note we extend this approach to also include individual statistical constraints.

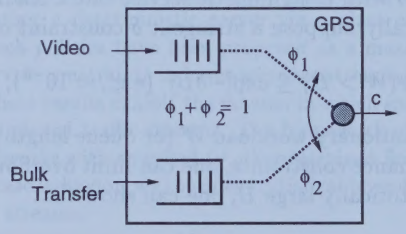


Figure 1: Segregating traffic and sharing capacity.

Figure 1 shows a simple example in which a video stream and a stream corresponding to bulk data transfer share a link with capacity c. The ϕ_1 and ϕ_2 roughly correspond to the fraction of bandwidth allocated to each stream, see [18] and §3 for details. A real-time video stream with stringent delay requirements might be assigned a large ϕ_1, however, when no video packets are queued the whole capacity can be devoted to transmitting data in the second buffer. In the worst case the GPS policy guarantees a backlogged buffer i a minimal bandwidth $\phi_i c$.

In this paper we find rough bandwidth requirements for such systems when each buffer must satisfy an overflow constraint. These in turn indicate how weighting parameters might be selected as well as appropriate call admittance and routing rules based on individual user requirements.

In §2 we give a brief overview of the large deviation results used in our analysis. In §3, we give a succinct description of GPS service policies, and state and discuss our

main result. Finally we conclude in §4 offering some further comments on performance issues in ATM.

2 Background - Large Deviations

Since our arguments are based on large deviations results, we begin with a very brief review. For a complete reference on the subject see Dembo et al. [7]. Consider the distributions $\{\mu_n\}$ of the partial sums $n^{-1}S_n = n^{-1}\sum_{i=1}^{n} A_n$, of a sequence of real-valued random variables $\{A_n\}$. We say that $\{A_n\}$ satisfies a large deviation principle (LDP) with *good rate function* $I(\cdot)$ if for every closed set F and open set G,

$$\limsup_{n\to\infty} \tfrac{1}{n}\log \mu_n(F) \leq -\inf_{x\in F} I(x) \text{ and}$$
$$\liminf_{n\to\infty} \tfrac{1}{n}\log \mu_n(G) \geq -\inf_{x\in G} I(x),$$

where $\{x : I(x) \leq \alpha\}$ is compact for $\alpha < \infty$. For example, if the sequence is ergodic with mean m and satisfies an LDP, then the probability that the empirical mean over a long time interval $[0, n]$ is greater than α is approximately

$$\mathbb{P}(\frac{S_n}{n} \geq \alpha) \approx \exp[-nI(\alpha)],$$

up to a logarithmic equivalence. Below we briefly discuss when such bounds do indeed hold.

The Gärtner-Ellis Theorem establishes the existence of an LDP with convex good rate function for a large class of sources. It requires that the limits $\Lambda(\theta) \triangleq \lim_{n\to\infty} \frac{1}{n}\log \mathbb{E}\exp[\theta S_n]$ exist (possibly infinite) for all $\theta \in \mathbb{R}$, in addition to two important but technical assumptions [7]. With these conditions in place, an LDP holds with a good rate function given by the convex dual, $\Lambda^*(\cdot)$, of $\Lambda(\cdot)$:

$$\Lambda^*(x) = \sup_{\theta}[\theta x - \Lambda(\theta)].$$

This result applies to i.i.d. sequences with $\mathbb{E}e^{\theta A_1} < \infty$ for all θ corresponding to the original large deviation estimate of Cramér. LDPs can also be found for sequences with weak dependencies; e.g., coordinate functions of Markov processes and mixing sequences [7].

3 Bandwidth Requirements and GPS

Consider a collection of traffic streams, J, sharing multiple segregated buffers, $i = 1 \ldots N$, generalizing the example in Figure 1. A particular buffer could handle either a single stream, or streams of the same type; e.g., several audio sessions. We will assume that the input traffic streams have known effective bandwidths. The total output rate, c, is shared among the buffers according to a GPS service policy; see [18, 19] for details. This has the advantage of guaranteeing minimal service rates to each of the buffers according to load sharing parameters $\{\phi_i\}_{i=1}^{N}$ preselected in network software. In particular, the

GPS policy guarantees that for any interval of time, say $[-n, 0)$, for which a buffer i has work to do, the cumulative departures from that buffer during the interval, $S_n^{D^i}$, satisfy

$$\frac{S_n^{D^i}}{S_n^{D^j}} \geq \frac{\phi_i}{\phi_j} \; \forall j \neq i. \tag{5}$$

This in turn implies that a buffer with a backlog is *guaranteed* a minimum output rate, in particular

$$S_n^{D^i} \geq n \frac{\phi_i}{\sum_{j \in J} \phi_j} c \triangleq n c_i.$$

This fractional guaranteed bandwidth is a worst case estimate. As noted in the introduction, a buffer i may get a much larger proportion of the total capacity of the link. For example, if all other buffers are idle, buffer i will see a service rate amounting to the total capacity of the output link c. When "spare capacity" is available, it is shared equitably among buffers requiring service, that is, it is shared in proportion to the respective weighting factors.

3.1 Large Buffer Asymptotics for Two-Buffer GPS

For simplicity, we present the case where there are only two buffers sharing a link with capacity c. Without loss of generality, the weighting parameters are normalized (i.e., $\phi_1 + \phi_2 = 1$) and we consider the distribution at buffer 1. Extensions to multiple streams and buffers are discussed in §3.2. In our analysis we use discrete-time *fluid* models for traffic streams; these are in turn related to slotted packet streams in §3.3.

Theorem 3.1 *Let $\{A_n^j\}_{n=-\infty}^{\infty}$, $j = 1, 2$ be stationary ergodic arrival processes with $\mathbb{E}[A_n^1 + A_n^2] < c$. Assume the streams are independent and satisfy an LDP (for example, assume they satisfy the Gärtner-Ellis Theorem) and that the limits*

$$\Lambda_j(\theta) = \lim_{n \to \infty} \frac{1}{n} \log \mathbb{E} \exp[\theta \sum_{i=1}^{n} A_i^j] < \infty, \text{ for all } \theta, \tag{6}$$

with $\Lambda_j^*(\cdot)$ strictly convex.[2] *Consider the queue length processes generated when the GPS policy is in effect:*

$$Q_{n+1}^1 = Q_n^1 + A_n^1 - D_n^1 \quad \text{and} \quad Q_{n+1}^2 = Q_n^2 + A_n^2 - D_n^2,$$

where the departures, D_n^1 and D_n^2, from the two buffers are consistent with Eq. 5. Then the stationary queue length for the first buffer, denoted by the random variable Q^1, satisfies

$$\lim_{B \to \infty} \frac{1}{B} \log \mathbb{P}(Q^1 > B) = - \inf_{\alpha_1 + \alpha_2 \wedge \phi_2 c > c} \frac{\Lambda_1^*(\alpha_1) + \Lambda_2^*(\alpha_2)}{\alpha_1 + \alpha_2 \wedge \phi_2 c - c}.$$

Alternatively (c.f., Section 3.2), for $\delta > 0$,

$$\alpha_1(\delta) + \alpha_2^D(\delta, \phi_2 c) \leq c \Longleftrightarrow \lim_{B \to \infty} \frac{1}{B} \log \mathbb{P}(Q^1 > B) \leq -\delta \tag{7}$$

[2] Weaker conditions are possible; see [5] for a case where weaker conditions were used.

where $\alpha_2^D(\delta, \phi_2 c)$ *defined below depends on bandwidth guaranteed to the second buffer. Indeed if* $\mathbb{E}A_n^2 < \phi_2 c$ *we have*

$$\alpha_2^D(\delta, \phi_2 c) := \begin{cases} \alpha_2(\delta) & \text{if } \alpha_2^*(\theta) < \phi_2 c \\ \phi_2 c - \Lambda_2^*(\phi_2 c)/\delta & \text{otherwise} \end{cases}$$

where $\alpha_2^*(\delta) := \arg\sup_\alpha \{\alpha\delta - \Lambda_2^*(\alpha)\}$ *and if* $\mathbb{E}A_n^2 \geq \phi_2 c$ *then* $\alpha_2^D(\delta, \phi_2 c) = \phi_2 c$.

Proof: We will give the proof of the lower bound only. The proof of the upper bound is obtained by a combination of the ideas in [2] and [3]; see [4].

We study the stationary distribution for the queue length in buffer 1 by way of Loynes's construction [17]. We set the queues empty at time $-m$ and consider the distribution induced at time 0 as the empty starting point, $-m$, is moved into the remote past. More specifically, let

$$\begin{aligned} Q_n^{1,m} &= 0 \text{ for } n \leq -m, \\ Q_{n+1}^{1,m} &= Q_n^{1,m} + A_n^1 - D_n^{1,m} \text{ for } n \geq -m, \\ &= [Q_n^{1,m} + A_n^1 - c + D_n^{2,m}]^+, \end{aligned}$$

where the last equality uses the fact that GPS is work conserving service policy and can be seen to correspond to Lindley dynamics. A similar set of equations defines the dynamics for the second queue. Note that the departures $D^{1,m}$ and $D^{2,m}$ have been indexed by m to indicate the fact that the departure process depends on the starting point m. The stability condition, $\mathbb{E}[A_n^1 + A_n^2] < c$, guarantees the existence of a stationary distribution. Following Loynes's argument we find that the distribution of $Q_0^{1,m}$ increases monotonically to that of Q^1. Define $S_0^{A^1} = 0$ and $S_n^{A^1} = \sum_{i=-n}^{-1} A_i^1$ for $n \geq 1$, and define $S_n^{A^2}$, $S_n^{D^{1,m}}$ and $S_n^{D^{2,m}}$ similarly. The distribution of $Q_0^{1,m}$ is given by that of a reflected random walk, which in turn has the following form [1, page 80]:

$$\begin{aligned} Q_0^{1,m} &= \max_{0 \leq n \leq m} S_n^{A^1} - (nc - S_n^{D^{2,m}}) \\ &= \max_{0 \leq n \leq m} S_n^{A^1} - (nc - S_n^{D^{2,m}}) \vee n\phi_1 c \\ &= \max_{0 \leq n \leq m} S_n^{A^1} - (nc - S_n^{D^{2,m}}) \vee (nc - n\phi_2 c) \\ &= \max_{0 \leq n \leq m} S_n^{A^1} + S_n^{D^{2,m}} \wedge n\phi_2 c - nc. \end{aligned} \tag{8}$$

Here $\wedge$ and $\vee$ denote respectively min and max. The second equality above follows from the fact that the n achieving the maximum on the right hand side corresponds to a non-idling busy period, $[-n, 0)$, for which the minimal guaranteed bandwidth property of GPS must hold; whence the additional max term does not affect the the right hand side. The third step uses the fact that $\phi_1 + \phi_2 = 1$.

To obtain the lower bound, notice that since the distribution of $Q_0^{1,m}$ increases monotonically to that of Q^1,

$$\begin{aligned} \mathbb{P}(Q^1 > B) &\geq \mathbb{P}(Q_0^{1,m} > B) \\ &\geq \mathbb{P}(S_m^{A^1} + S_m^{D^{2,m}} \wedge m\phi_2 c - mc > B) \\ &\geq \mathbb{P}(S_m^{A^1} + S_m^{A^2} \wedge m\phi_2 c - mc > B) \end{aligned}$$

where $S_m^{A^2} \geq S_m^{D^{2,m}}$ because we start both queues empty. By letting $m = \lfloor B/\beta \rfloor$ for $\beta > 0$, we have

$$\begin{aligned}
&\liminf_{B\to\infty} \frac{1}{B} \log \mathbb{P}(Q^1 > B) \\
&\geq \frac{1}{\beta} \liminf_{m\to\infty} \frac{1}{m} \log \mathbb{P}(S_m^{A^1} + S_m^{A^2} \wedge m\phi_2 c - mc > m\beta) \\
&\geq \frac{1}{\beta} \liminf_{m\to\infty} \frac{1}{m} \log \mathbb{P}(\frac{S_m^{A^1}}{m} + \frac{S_m^{A^2}}{m} \wedge \phi_2 c - c > \beta) \\
&\geq \frac{1}{\beta} \liminf_{m\to\infty} \frac{1}{m} \log \mathbb{P}(f(\frac{S_m^{A^1}}{m}, \frac{S_m^{A^2}}{m}) > \beta + c)
\end{aligned}$$

where $f(\alpha_1, \alpha_2) = \alpha_1 + \alpha_2 \wedge \phi_2 c$ which is a continuous function. Using the contraction principle we obtain a large deviations lower bound for this term (see [7, page10]):

$$\begin{aligned}
&\liminf_{B\to\infty} \frac{1}{B} \log \mathbb{P}(Q^1 > B) \\
&\geq -\frac{1}{\beta} \inf_{\{(\alpha_1,\alpha_2):\ f(\alpha_1,\alpha_2)=\beta+c\}} \Lambda_1^*(\alpha_1) + \Lambda_2^*(\alpha_2).
\end{aligned}$$

In addition, we find the tightest such bound by optimizing over $\beta > 0$:

$$\begin{aligned}
&\liminf_{B\to\infty} \frac{1}{B} \log \mathbb{P}(Q^1 > B) \\
&\geq -\inf_{\beta>0} \inf_{\{(\alpha_1,\alpha_2):\ f(\alpha_1,\alpha_2)=\beta+c\}} \frac{\Lambda_1^*(\alpha_1) + \Lambda_2^*(\alpha_2)}{\beta} \\
&= -\inf_{\alpha_1+\alpha_2\wedge\phi_2 c>c} \frac{\Lambda_1^*(\alpha_1) + \Lambda_2^*(\alpha_2)}{\alpha_1 + \alpha_2 \wedge \phi_2 c - c}.
\end{aligned}$$

By using the strict convexity of the rate functions and the fact that the rate function is zero at the mean rate we can show that the solution to this optimization problem is δ precisely when $\alpha_1(\delta) + \alpha_2^D(\delta, \phi_2 c) = c$. Note that in order to do so we will have to handle several cases in particular depending on whether the mean rate entering second buffer exceeds its guaranteed bandwidth. □

3.2 Interpretation and Multiple Buffers

The result for the two buffer two stream case studied in §3.1 has a clean interpretation. In order to guarantee an overflow δ-constraint on buffer 1 (Eq. 7) we require that $\alpha_1(\delta) + \alpha_2^D(\delta, \phi_2 c) < c$. In turn $\alpha_2^D(\delta, \phi_2 c)$ can assume one of two values. If $\alpha_2^*(\delta) < \phi_2 c$, then the usual effective bandwidth constraint is obtained (i.e., $\alpha_1(\delta) + \alpha_2(\delta) < c$); otherwise, the *firewall* set up by the GPS service policy comes into play and the requirement depends on the service rate and weighting parameters becoming $\alpha_1(\delta) < \phi_1 c + \Lambda_2^*(\phi_2 c)/\delta$. In the first case we call the system decoupled and say that, with respect to the desired constraint, it behaves as if the resources were pooled together. In the second case we say that the second buffer is saturated and that segregation is taking place.

For multiple buffers and *independent* stationary streams, the results have a similar structure. As an example suppose that there are three buffers sharing a total

capacity c via a GPS service policy with weighting factors ϕ_1, ϕ_2 and ϕ_3. Let $\alpha_i(\delta)$, $\alpha_i^*(\delta)$ and $\alpha_i^D(\delta, \phi_i c)$ respectively denote the effective, decoupling and output bandwidths for each buffer defined in the previous sections. Say buffer 1 has a real-time video stream, buffer 2 is shared by N audio streams, and buffer 3 carries traffic corresponding a single LAN-to-LAN connection. An overflow δ_1-constraint on buffer 1 is guaranteed if $\alpha_1(\delta_1) + \alpha_2^D(\delta_1, \phi_2 c) + \alpha_3^D(\delta_1, \phi_3 c) < c$. Note $\alpha_2^D(\delta_1, \phi_2 c)$ corresponds to the aggregate effect of N identical audio streams; a simple argument shows that $\alpha_2^D(\delta_1, \phi_2 c) = N\alpha_{audio}^D(\delta_1, \phi_2 c/N)$. As in the two buffer case, several regimes exist in which all the buffers are decoupled or some are saturated depending on the decoupling bandwidths and the weighting parameters. Similar constraints are obtained for the other buffers where we might ensure other QoS constraints; e.g., δ_2 and δ_3.

Note we have assumed traffic streams are independent, stationary ergodic, and satisfying an LDP. This includes most models used in modeling ATM traffic, such as Markov fluid models, and in turn the effective and decoupling bandwidths can be explicitly computed for most of the models of interest.

3.3 From GPS to Packetized GPS (PGPS)

In the previous section, the departure processes D^j were real valued; i.e., the buffers described above are discrete-time "fluid" models of ATM buffers. ATM traffic consists of streams of 53-byte packets called cells. In ATM, the buffers' server cannot handle two cells simultaneously. Consequently, GPS cannot be implemented in an ATM network.

A packetized version of GPS (PGPS) has been proposed; in [8], PGPS is called "weighted fair queueing." In PGPS, the departure times of the cells under GPS are calculated. The cells under PGPS are served in the order of these calculated departure times. A consequence of Theorem 2 in [18] is

$$\alpha_1(\delta) + \alpha_2^D(\delta, \phi_2 c) < c \;\Rightarrow\; \lim_{B\to\infty} \frac{1}{B} \log \mathbb{P}(\hat{Q}^1 > B) \leq \delta,$$

where $\hat{Q}^1$ is the steady-state queue length under PGPS.

4 Summary

In summary, we have obtained effective bandwidth results for GPS and PGPS buffering systems, in the context of ATM-based B-ISDN networks, allowing for a circuit-switched style admission control and routing for multiple QoS. An important aspect of this work is that it allows integration of both worst case bounds of [18] with more efficient statistical QoS guarantees.

In this paper, we have concentrated on the single node scenario. We are currently attempting to characterize the effective bandwidth of the output process, when the service policy is GPS, in order to justify the extension to networks. We note however that in general the stability of such a network in not obvious and point the reader to [23] for further discussion.

References

[1] S. Asmussen. *Applied Probability and Queues.* John Wiley & Sons, 1987.

[2] C.S. Chang. Stability, queue length and delay of deterministic and stochastic queueing networks. *IEEE Automatic Control*, 39(5):913–931, May 1994.

[3] G. de Veciana, C. Courcoubetis, and J. Walrand. Decoupling bandwidths for networks: A decomposition approach to resource management for networks. *IEEE Infocom Proceedings '94, also submitted to IEEE/ACM Trans. Networking*, 1993.

[4] G. de Veciana and G. Kesidis. Bandwidth allocation for multiple qualities of service using generalized processor sharing. Technical Report SCC-94-01, U.T. Austin, ECE Department.

[5] G. de Veciana and J. Walrand. Traffic shaping for ATM networks: Asymptotic analysis and simulations. *Submitted to IEEE/ACM Trans. Networking*, 1993.

[6] G. de Veciana and J. Walrand. Effective bandwidths: Call admission, traffic policing and filtering for ATM networks. *To appear Queueing Systems*, 1994.

[7] A. Dembo and O. Zeitouni. *Large Deviations Techniques and Applications.* Jones & Bartlett, Boston, 1992.

[8] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. *Internet. Res. and Exper.*, 1, 1990.

[9] B.T. Doshi. Deterministic rule based traffic descriptors for broadband ISDN: Worst case behavior and concept equivalent bandwidth. In *GLOBECOM*, pages 1759–1764, 1993.

[10] A. E. Eckberg. B-ISDN/ATM traffic and congestion control. *IEEE Network*, Sept. 1992.

[11] A. I. Elwalid and D. Mitra. Effective bandwidth of general Markovian traffic sources and admission control of high speed networks. *IEEE/ACM Trans. Networking*, 1(4), June 1993.

[12] R.J. Gibbens and P.J. Hunt. Effective bandwidths for the multi-type UAS channel. *Queueing Systems*, 9:17–28, 1991.

[13] R. Guérin, H. Ahmadi, and M. Naghshineh. Equivalent capacity and its application to bandwidth allocation in high-speed networks. *IEEE JSAC*, 9:968–981, 1991.

[14] R. Guérin and L. Gün. A unified approach to bandwidth allocation and access control in fast packet-switched networks. In *IEEE Infocom Proceedings*, 1992.

[15] F.P. Kelly. Effective bandwidths at multi-class queues. *Queueing Systems*, 9:5–16, 1991.

[16] G. Kesidis, J. Walrand, and C.S. Chang. Effective bandwidths for multiclass Markov fluids and other ATM sources. *IEEE/ACM Trans. Networking.*, 1(4), August 1993.

[17] R.M. Loynes. The stability of a queue with non-independent inter-arrivals and service times. *Proc. Camb. Phil. Soc.*, 58:497–520, 1962.

[18] A.K. Parekh and R.G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The single node case. *IEEE/ACM Trans. Networking*, 1(3), June 1993.

[19] A.K. Parekh and R.G. Gallager. A generalized processor sharing approach to flow control: The multiple node case. *IEEE/ACM Trans. Networking*, July 1994.

[20] K. Sriram. Methodologies for bandwidth allocation, transmission scheduling, and congestion avoidance in broadband ATM networks. *Computer Networks and ISDN Systems*, 26:43–59, 1993.

[21] M. Wernik, O. Aboul-Magd, and H. Gilbert. Traffic management for B-ISDN services. *IEEE Network*, Sept. 1992.

[22] W. Whitt. Tail probabilities with statistical multiplexing and effective bandwidths in multi-class queues. *Telecommunication Systems*, 2:71–107, 1993.

[23] O. Yaron and M. Sidi. Generalized processor sharing networks with exponentially bounded burstiness arrivals. In *Infocom*, 1994.

Quick Detection of Changes in Traffic Statistics: Application to Variable Rate Compression*

Ivy Hsu and Jean Walrand
Department of EECS, University of California, Berkeley, CA 94720

Abstract

In many communication applications, sources generate data at a variable rate. This variation can be captured by a two-level model. At a larger time scale, the source alternates between a finite number of models at the higher level, while each model represents the local fluctuation at the smaller time scale. Typically the data goes through a lossy compression at the encoder, and then is entered into a buffer before being transmitted over a fixed-rate or a regulated-rate (e.g., Leaky Bucket) channel. Since the buffer size is finite, the choice of compression parameters must balance the tradeoff between the distortion due to lossy compression and the distortion caused by buffer overflow. As the optimal compression parameters may be different for each individual model, a method is needed to detect changes from one model to another.

We show via examples that for many common traffic models, the problem of detecting changes in traffic statistics can be reduced to the following general change point detection problem: Let $A_1, A_2, \ldots$ be a series of independent observations such that $A_1, A_2, \ldots, A_{m-1}$ are i.i.d. distributed according to an unknown distribution F, and $A_m, A_{m+1}, \ldots$ are i.i.d. distributed according to another unknown distribution G. m is the unknown change point. The objective is to determine that a change has occurred as soon as possible, while maintaining a low rate of false alarms. In this paper we propose a sequential nonparametric change point detection algorithm using Kolmogorov-Smirnov statistics, and demonstrate that it is *asymptotically optimal*, in that under the constraint $E[\text{time until a false alarm}] \geq T$, our algorithm is such that $E[\text{detection delay}] = O(\log(T))$, as $T \to \infty$. This property is known to be the best that can be achieved. Furthermore, unlike other nonparametric schemes, the performance of the algorithm is independent of the distributions F and G.

1 Introduction

Consider communication applications such as MPEG video, in which the output stream generated by the source has a variable rate due to statistical variations in the data. The output stream is released onto a channel at a constant rate or at a regulated rate that allows some burstiness. Fixed bandwidth radio channels for wireless communication are

*This research is supported in part by a grant of the Networking and Communications Research Program of the National Science Foundation.

an example of the former. The Leaky Bucket scheme proposed for the Asynchronous Transfer Mode (ATM) networks represents the latter. In both cases, unless the transmission rate exceeds the peak rate of the source, it is necessary to place a buffer between the source encoder and the channel to store data when they are generated faster than what the channel can transmit.

First suppose that the source can be characterized by a time invariant model. Then for any given compression scheme, the quantities of interest are its mean rate and mean-square distortion measure for the given source model. Since the buffer size is finite, one must consider the tradeoff between this distortion measure and the distortion caused by lost data when the buffer overflows. A finer compression reduces distortion, but generates more data and thus leads to higher overflow probability.

Define the distortion-rate curve as the minimum distortion achievable by any scheme under an average rate constraint. Tse et al. [8] suggest that if the distortion-rate curve is concave at the channel rate R_c, then one can achieve a lower distortion by time-sharing between two compression schemes. Define $D_T(R_c)$ as the minimum distortion achievable by any time-sharing scheme under an average rate constraint R_c. Then the objective is for the buffer overflow probability and the steady state average distortion when the buffer is not full to approach zero and $D_T(R_c)$, respectively, as the buffer size becomes large. The control scheme proposed in [8] places a threshold at the half point of the buffer, and uses a coarse compression with average rate $R_1 < R_c$ when the buffer occupancy exceeds the threshold and a fine compression with rate $R_2 > R_c$ otherwise. Under this scheme, both the overflow probability and the average distortion approach their asymptotic values exponentially fast as the buffer size goes to infinity.

Under a more refined assumption, the source can be viewed to be alternating between a finite number of models, at a time scale much larger than the time scale of fluctuation captured by each individual model. For example, a video sequence can contain active and quiet scenes, with a mean scene duration of several seconds. The active scenes in general cannot be compressed as well as the quiet ones. Therefore the optimal parameters of the compression scheme would be different in each model.

Thus the encoder must be able to respond quickly when a change in the source model occurs and identify the new model. One approach is to start with a fixed set of source models and parameters for which the control parameters are known, and conduct a *parametric* detection, i.e., compare the incoming traffic with all existing models, and determine the model that is closest. Then apply the corresponding control scheme. This approach, although simple in nature, has a limited accuracy and requires extensive analyses of all possible traffic statistics. A more general approach is to make no prior assumption about source models, and adaptively determine the control parameters to use for each model. Thus the algorithm for change point detection must be *nonparametric.* The system maintains a library of past source statistics and corresponding control parameters. Once a change is detected, the new statistics are compared with entries in the library in order to determine if it belongs to one of the existing models. If so, the control parameters associated with the identified model are used. Otherwise, a new entry is created in the library, and the control parameters for the closest model are selected as its starting point. In both cases, these parameters are then adjusted over time, as the measured distortion is compared with the target.

As an example, Fig. 1 illustrates the block diagram of a possible control scheme

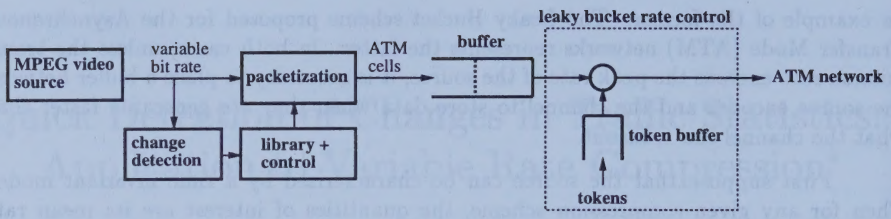


Figure 1: Transporting a variable rate MPEG sequence over an ATM network with rate constraint.

for variable rate video applications. The packetization block segments data generated by the video source into ATM cells of two priorities, in a way that incorporates the multiresolution principle: decoding with the high priority cells alone results in a coarser quality of video, while decoding the combined priorities results in a finer quality. The proportion of the two priorities is controlled based on the source statistics as mentioned above. The buffer operates under a *partial buffer sharing* mechanism [5], in which only high priority cells can enter the buffer when the occupancy is above a threshold. The data is emptied from the buffer through a leaky bucket before entering an ATM network.

In this paper we address the issue of detecting changes in traffic statistics. The general change point (or disorder) detection problem has undergone much investigation since 1950s. For the application at hand we are interested in a sequential nonparametric detection method. The main criterion of an effective sequential scheme is that it should respond to changes as soon as possible, while maintaining a low frequency of false alarms. In addition, it is desirable for the decision rule to be *distribution-free*, i.e., its effectiveness should not vary for different pairs of statistics before and after the change point. Finally, from the implementation point of view, the scheme should require only a bounded number of operations and memory at each decision point.

Several sequential nonparametric detection methods have been proposed previously. For a comprehensive review, see [2]. Most of these results are concentrated on the case in which the observed sequence is a concatenation of two i.i.d. random sequences, with some limited extension to n-dependent random sequences. Their common drawback, however, is the lack of the distribution-free property. For example, the detection scheme in [1] compares the empirical means before and after each decision point. Consequently the false alarm frequency depends on the variance of the distribution before the change.

In §2, we present a general detection algorithm based on comparisons of empirical distributions. We show via examples how this algorithm can be applied in the context of detecting changes in incoming traffic for many common models. The statistic used in this case is the Kolmogorov-Smirnov distance, which has the desirable property of being distribution-free [4]. This statistic has been used in [3] for the *a posteriori* change point problem, in which the emphasis is on accurate identification of the most likely point of change within a given set of observations. §3 presents the main theorem of the paper. We show that the ratio of the expected delay of a correct detection to the logarithm of the expected time until a false alarm is upper bounded by a constant, as the memory size of the algorithm increases. This logarithmic relation implies that the proposed scheme belongs to the class of asymptotically optimal algorithms, as indicated in [1, 6]. We present some preliminary numerical results in §4 and conclude in §5.

2 Change Point Detection Algorithm

2.1 Assumptions

Let $\{A_n\}$ be the observed sequence and m the unknown change point. Denote $1_{(\cdot)}$ as the indicator function. Assume that $A_n = \eta_n 1_{(n<m)} + \xi_n 1_{(n\geq m)}, n \geq 1$, where η_n's (resp. ξ_n's) are i.i.d. random variables with a continuous distribution F (resp. G). Furthermore, assume that there exists a constant $\varepsilon > 0$ such that the set $\mathcal{A} = \{x \in \mathbf{R} : |F(x) - G(x)| \geq \varepsilon\}$ satisfies either $\int_{\mathcal{A}} dF(x) > 0$ or $\int_{\mathcal{A}} dG(x) > 0$.

At the first glance the i.i.d. assumption may appear restrictive. But as the following examples demonstrate, A_n can be derived in many complex traffic models:

Example 1 Markov Modulated Sources *One common model for bursty sources is the Markov modulated fluid, in which the instantaneous arrival rate from a source depends on the state of an underlying Markov chain. Consider, for example, an ON-OFF process, in which the Markov chain alternates between two states, and the source generates data at a constant rate when the chain is ON and no data otherwise. The source statistics can differ when there is a change in the transition rates of the Markov chain, or in the output rate, or both. Choose the regenerative points t_n as the successive times at which the process returns to the OFF state. Then define $A_n \equiv$ (total output in $[t_n, t_{n+1})$)$/(t_{n+1} - t_n)$.*

Example 2 Periodic Sources *Periodic sources arise when input data is encoded at fixed time intervals (e.g., PCM of voice, frame-by-frame encoding of video). Thus A_n can be taken as the data generated over any constant multiple of a period. §4 gives a more complex example in the case of MPEG video, where the source can be treated as a superposition of several periodic streams.*

2.2 Algorithm

The algorithm is characterized by three parameters: the memory size N, the windowing parameter a, and the threshold c. At each time n, only the last N observations need to be stored in memory. The parameter a is a constant in $(0, \frac{1}{2}]$. For simplicity in notation, let a be chosen such that $aN \in \mathcal{N}$. Given the memory of N points, a defines a window such that each k in $\{aN, aN+1, \ldots, (1-a)N\}$ is a candidate change point. Define, for $n \geq N$ and $k \in [aN, (1-a)N]$,

$$D_N(k,n) = \max_{n-N+1 \leq i \leq n} \left| \frac{1}{k} \sum_{j=n-N+1}^{n-N+k} 1_{(A_j \leq A_i)} - \frac{1}{N-k} \sum_{j=n-N+k+1}^{n} 1_{(A_j \leq A_i)} \right|, \quad (1)$$

$$E_N(n) = \max_{aN \leq k \leq (1-a)N} D_N(k,n). \quad (2)$$

Let the decision rule be $d_N(n) = 1_{(E_N(n) > c)}$, where $0 < c < 1$. The detection time is then $\tau(N) = \min\{n \geq N : d_N(n) = 1\}$.

The algorithm can be interpreted as follows: $D_N(k,n)$ is the Kolmogorov-Smirnov distance, namely the maximum absolute difference, between the empirical distributions

of two sequences, $(A_{n-N+1}, \ldots, A_{n-N+k})$ and $(A_{n-N+k-1}, \ldots, A_n)$. For each $aN \leq k \leq (1-a)N$, we calculate the Kolmogorov-Smirnov distance and declare that a change has occurred if at any point k the distance exceeds the threshold c.

3 Theorem

Let $P_\infty(\cdot)$ (resp. $E_\infty(\cdot)$) represent the conditional probability (resp. expectation) given that no change point ever occurs. Similarly, let $P_m(\cdot)$ (resp. $E_m(\cdot)$) be the conditional probability (resp. expectation) given that change occurs at time m. As mentioned earlier, we are interested in two quantities:

(1) $E_\infty[\tau(N) - N] \equiv$ expected time until a false alarm

(2) $E_m[(\tau(N) - m)|\tau(N) \geq m] \equiv$ expected detection delay

The objective is to detect the change as fast as possible, while maintaining a large expected time between false alarms. The following theorem relates the two quantities under the above algorithm.

Theorem 1 *Given that*

$$\varepsilon \geq c + \left(\frac{-1}{2a(1-a)} \log(1 - c/2) \right)^{1/2}, \tag{3}$$

$$\limsup_{N \to \infty} \sup_{m \geq N} \frac{E_m[(\tau(N) - m)|\tau(N) \geq m]}{\log E_\infty[\tau(N) - N]} \leq \textit{Constant}. \tag{4}$$

It is pointed out in [1, 6] that for the problem of minimizing $\sup_m E_m[(\tau(N) - m)|\tau(N) \geq m]$ under the constraint $E_\infty[\tau(N) - N] \geq T$, the optimal sequential detection rule admits an asymptotic behavior $\sup_m E_m[(\tau(N) - m)|\tau(N) \geq m] = O(\log(T))$, as $T \to \infty$. Thus by virtue of theorem 1, the proposed algorithm is asymptotically optimal.

Theorem 1 follows from Lemmas 2 and 3 below. We first derive an upper bound on the probability of false alarm at time $N + k$, for all $k \geq 0$.

Lemma 1

$$P_\infty(\tau(N) - N = k) \leq L_1(N) N e^{-L_2(N)N}, \quad \forall k \geq 0, \tag{5}$$

where $L_1(N)$ and $L_2(N)$ are defined below.

Proof Define $\beta_N(n) = P_\infty(d_N(n) = 1)$ for all $n \geq N$, then $P_\infty(\tau(N) - N = k) \leq \beta_N(N + k)$. Given that no change takes place, the samples are i.i.d. $\sim F$. Thus $\beta_N(n) \equiv \beta_N(N)$, for all $n \geq N$. Then

$$\beta_N(N) = P_\infty \left(\max_{aN \leq k \leq (1-a)N} D_N(k, N) > c \right) \leq \sum_{aN \leq k \leq (1-a)N} P_\infty(D_N(k, N) > c). \tag{6}$$

$$
\begin{aligned}
& P_\infty(D_N(k,N) > c) \\
\overset{(a)}{\leq} \; & P_\infty\left(\max_{1\leq i\leq N} |\frac{1}{k}\sum_{j=1}^{k} 1_{(A_j\leq A_i)} - F(A_i)| + |\frac{1}{N-k}\sum_{j=k+1}^{N} 1_{(A_j\leq A_i)} - F(A_i)| > c\right) \\
\overset{(b)}{\leq} \; & P_\infty\left(\max_{1\leq i\leq N} |\frac{1}{k}\sum_{j=1}^{k} 1_{(A_j\leq A_i)} - F(A_i)| + \max_{1\leq i\leq N} |\frac{1}{N-k}\sum_{j=k+1}^{N} 1_{(A_j\leq A_i)} - F(A_i)| > c\right) \\
\overset{(c)}{\leq} \; & P_\infty\left(\sup_x |\frac{1}{k}\sum_{j=1}^{k} 1_{(A_j\leq x)} - F(x)| > c/2\right) + \\
+ \; & P_\infty\left(\sup_x |\frac{1}{N-k}\sum_{j=k+1}^{N} 1_{(A_j\leq x)} - F(x)| > c/2\right);
\end{aligned} \tag{7}
$$

where (a) is due to the triangular inequality, and (b) follows from the fact that the maximum of the sum is less than or equal to the sum of the maxima. (c) results from $P(X+Y>c) \leq P(X>c/2 \text{ or } Y>c/2) \leq P(X>c/2)+P(Y>c/2)$, and that the maximum distance between an empirical distribution and the actual distribution at the sample points is upper bounded by the supremum distance.

$D_k \equiv \sup_x |\frac{1}{k}\sum_{j=1}^{k} 1_{(A_j\leq x)} - F(x)|$ is the Kolmogorov-Smirnov one-sample statistic based on k sample points. It can be expressed as the maximum of two non-negative one-sided Kolmogorov-Smirnov statistics D_k^+ and D_k^-, defined as

$$
D_k^+ \equiv \sup_x \left[\frac{1}{k}\sum_{j=1}^{k} 1_{(A_j\leq x)} - F(x)\right] = \max_{1\leq i\leq k}\left[\frac{i}{k} - F(A_{(i)}^k)\right], \tag{8}
$$

and

$$
D_k^- \equiv \sup_x \left[F(x) - \frac{1}{k}\sum_{j=1}^{k} 1_{(A_j\leq x)}\right] = \max_{1\leq i\leq k}\left[F(A_{(i)}^k) - \frac{i-1}{k}\right], \tag{9}
$$

where $A_{(1)}^k < A_{(2)}^k < \cdots < A_{(k)}^k$ are the order statistics of $A_1, \ldots, A_k$. The superscript represents the total number of sample points.

Since F is continuous, by the probability-integral transformation theorem, given a random variable $A \sim F$, the random variable U generated by the transformation $U = F(A)$ has a Uniform(0, 1) distribution. Therefore, $U_{(i)}^k = F(A_{(i)}^k)$ is the i^{th} order statistic of k i.i.d. Uniform(0, 1) random variables. Thus for any fixed $\epsilon > 0$,

$$
P(\max_{1\leq i\leq k} \frac{i}{k} - F(A_{(i)}^k) > \epsilon) = P(\max_{1\leq i\leq k} \frac{i}{k} - U_{(i)}^k > \epsilon) \leq \sum_{i=1}^{k} P(U_{(i)}^k < \frac{i}{k} - \epsilon). \tag{10}
$$

Note that for $i/k < \epsilon$, $P(U_{(i)}^k < \frac{i}{k} - \epsilon) = 0$. For $i/k \geq \epsilon$, $P(U_{(i)}^k < \frac{i}{k} - \epsilon)$ is the probability that out of k i.i.d. Uniform(0, 1) random variables, at least i of them are less than $i/k - \epsilon$. Let $W_j \equiv 1(U_j < i/k - \epsilon)$. Then $P(U_{(i)}^k < \frac{i}{k} - \epsilon) = P(\sum_{j=1}^{k} W_j \geq i)$, which is the probability that the sample average is greater than i/k, while $E[W_1] = i/k - \epsilon$.

Define $\varphi(\theta) \equiv \log E e^{\theta W_1}$. Then by Markov inequality, $P(\sum_{j=1}^{k} W_j \geq \frac{i}{k}\cdot k) \leq \exp(-k\cdot\frac{i}{k}\theta)\exp(k\cdot\varphi(\theta))$. In particular, $P(\sum_{j=1}^{k} W_j \geq \frac{i}{k}\cdot k) \leq \exp(-k\sup_\theta[\frac{i}{k}\theta - \varphi(\theta)]) \equiv \exp(-kI_+(i,k,\epsilon))$, where

$$
I_+(i,k,\epsilon) = \begin{cases} \frac{i}{k}\log\left(\frac{i/k}{i/k-\epsilon}\right) + (1-\frac{i}{k})\log\left(\frac{1-i/k}{1-i/k+\epsilon}\right), & \text{for } \frac{i}{k} > \epsilon \\ \infty, & \text{for } \frac{i}{k} \leq \epsilon. \end{cases} \tag{11}
$$

Similarly, we can show that $P(U_{(i)}^k - \frac{i-1}{k} > \epsilon) \leq \exp(-kI_-(i,k,\epsilon))$, and

$$I_-(i,k,\epsilon) = \begin{cases} \frac{i-1}{k}\log\left(\frac{(i-1)/k}{(i-1)/k+\epsilon}\right) + (1-\frac{i-1}{k})\log\left(\frac{1-(i-1)/k}{1-(i-1)/k-\epsilon}\right), & \text{for } \frac{i-1}{k} < 1-\epsilon \\ \infty, & \text{for } \frac{i-1}{k} \geq 1-\epsilon. \end{cases} \tag{12}$$

Define $I_N(\epsilon) = \min_{aN \leq k \leq (1-a)N} \min_{1 \leq i \leq k} \min\{I_+(i,k,\epsilon), I_-(i,k,\epsilon)\}$. Then $P_\infty(D_k > c/2) \leq P(D_k^+ > c/2) + P(D_k^- > c/2) \leq 2k \cdot \exp(-kI_N(c/2))$. Substituting into Eqn. 6,

$$\begin{aligned} \beta_N(N) &\leq \sum_{aN \leq k \leq (1-a)N} P_\infty(D_k > c/2) + P_\infty(D_{N-k} > c/2) \\ &\leq \sum_{aN \leq k \leq (1-a)N} 2\left[k\exp(-kI_N(c/2)) + (N-k)\exp(-(N-k)I_N(c/2))\right] \\ &\leq \frac{4(1-a)}{1-\exp(-I_N(c/2))} N e^{-aI_N(c/2)N} \equiv L_1(N) N e^{-L_2(N)N}. \end{aligned} \tag{13}$$

Note that if $P_\infty(\tau(N) - N = j) \leq \beta$ for all $j \geq 0$, then $E_\infty[\tau(N) - N]$ is minimized if

$$P_\infty(\tau(N) - N = j) = \begin{cases} \beta, & j \leq j^* = \lfloor 1/\beta \rfloor, \\ 1 - j^*\beta, & j = j^* + 1, \\ 0, & \text{otherwise.} \end{cases}$$

Thus $E_\infty[\tau(N) - N] \geq \frac{1}{2\beta}(1 - o(1))$, where $o(1) \to 0$ as $\beta \to 0$. Since $\beta_N(N) \to 0$ as $N \to \infty$, applying lemma 1, we obtain a lower bound on the expected time until a false alarm:

Lemma 2

$$E_\infty[\tau(N) - N] \geq \frac{1 - \exp(-I_N(c/2))}{8(1-a)N} e^{aI_N(c/2)N}(1 - o(1)), \tag{14}$$

where $o(1) \to 0$ *as* $N \to \infty$.

Before giving the upper bound on the expected detection delay, we first state the Hoeffding's inequality:

Theorem 2 (Hoeffding's Inequality) *[7] Let* $Y_1, Y_2, \ldots, Y_N$ *be independent random variables with zero means and bounded ranges:* $a_i \leq Y_i \leq b_i$. *For each* $\eta > 0$,

$$P(|Y_1 + \cdots + Y_N| \geq \eta) \leq 2\exp\left[\frac{-2\eta^2}{\sum_{i=1}^N (b_i - a_i)^2}\right]. \tag{15}$$

Lemma 3 *Given condition 3,*

$$E_m[\tau(N) - m | \tau(N) \geq m] \leq N\left(1 + \frac{2}{1 - aN\beta_N}\right). \tag{16}$$

Proof Define $p_N \equiv P_m(d_N(m+(1-a)N-1)=0)$, and $q_N \equiv P_m(d_N(m+N)=1)$. Then the following inequality is given in [1]:

$$E_m[\tau(N)-m|\tau(N)\geq m] \leq N\left(1+\frac{p_N/q_N}{1-aN\beta_N}\right). \tag{17}$$

Denote $F_i(\cdot)$ (resp. $G_i(\cdot)$) as the empirical distribution function generated by i i.i.d. samples with distribution F (resp. G). Then

$$D_N(k,m+(1-a)N-1) = \sup_x |\frac{aN}{k}F_{aN}(x)+\frac{k-aN}{k}G_{k-aN}(x)-G_{N-k}(x)|. \tag{18}$$

Therefore, for any fixed $x_0 \in \mathcal{A}$,

$$p_N \leq P(\sup_x |F_{aN}(x)-G_{(1-a)N}(x)| \leq c) \leq P(|F_{aN}(x_0)-G_{(1-a)N}(x_0)| \leq c). \tag{19}$$

By the triangular inequality,

$$|F_{aN}(x_0)-G_{(1-a)N}(x_0)-F(x_0)+G(x_0)|+|F_{aN}(x_0)-G_{(1-a)N}(x_0)| \geq |F(x_0)-G(x_0)| > \varepsilon. \tag{20}$$

Since the condition implies that $\varepsilon \geq c$,

$$\begin{aligned} p_N &\leq P(|F_{aN}(x_0)-F(x_0)+G(x_0)-G_{(1-a)N}(x_0)| > \varepsilon - c) \\ &= P(|\frac{1}{a}\sum_{j=1}^{aN}[1_{(\eta_j \leq x_0)}-F(x_0)]+\frac{1}{1-a}\sum_{j=aN+1}^{N}[G(x_0)-1_{(\xi_j \leq x_0)}]| > N(\varepsilon-c)). \end{aligned} \tag{21}$$

Applying the Hoeffding's Inequality to Eqn. 21, we get

$$b_j = \begin{cases} \frac{1}{a}[1-F(x_0)], & \text{for } 1 \leq j \leq aN, \\ \frac{1}{1-a}[G(x_0)], & \text{for } aN+1 \leq j \leq N, \end{cases}$$

$$a_j = \begin{cases} \frac{1}{a}[-F(x_0)], & \text{for } 1 \leq j \leq aN, \\ \frac{1}{1-a}[G(x_0)-1], & \text{for } aN+1 \leq j \leq N. \end{cases}$$

Thus $\sum_{j=1}^{N}(b_j-a_j)^2 = aN/a^2+(1-a)N/(1-a)^2 = N/a(1-a)$, and $p_N \leq 2\exp[-2Na(1-a)(\varepsilon-c)^2]$.

On the other hand, we obtain the following lower bound for q_N:

$$\begin{aligned} q_N &\geq P(\sup_x |G_{aN}(x)-G(x)+G(x)-G_{(1-a)N}(x)| > c) \\ &\geq P(\sup_x(G_{aN}(x)-G(x))-\sup_x(G_{(1-a)N}(x)-G(x)) > c) \\ &\geq P(\sup_x(G_{aN}(x)-G(x)) > c/2)\cdot P(\sup_x(G_{(1-a)N}(x)-G(x)) < -c/2) \\ &= P(D_{aN}^{+} > c/2)\cdot P(D_{(1-a)N}^{-} > c/2) \\ &= P(\max_{1\leq i\leq aN}\frac{i}{aN}-U_{(i)}^{aN} > c/2)\cdot P(\max_{1\leq i\leq (1-a)N}U_{(i)}^{(1-a)N}-\frac{i-1}{(1-a)N} > c/2) \\ &\geq P(U_{(aN)}^{aN} < 1-c/2)\cdot P(U_{(1)}^{(1-a)N} > c/2) = (1-c/2)^N. \end{aligned} \tag{22}$$

$p_N \leq 2q_N$ for all N if $e^{-2a(1-a)(\varepsilon-c)^2} \leq 1-c/2$, or equivalently, if condition 3 holds.

Note that $I_N(c/2) \geq \inf_{x>c/2}[x\log\frac{x}{x-c/2}+(1-x)\log\frac{1-x}{1-x+c/2}], \forall N$, and thus Theorem 1 follows.

4 Numerical Example

In the variable-rate video coding schemes such as MPEG, the number of bits generated by each frame depends on both its information content and the encoding technique used. In MPEG, the choices of encoding modes are intraframe (I), predictive (P), and interpolative (B), in the order of increasing coding efficiency. A typical sequence consists of periodic interleaving of frames from each mode. For example, the sequence in our example has a period of p =15 frames in the following pattern: {I, B, B, P, B, B, P, B, B, P, B, B, P, B, B}.

One approach is to define A_n as the total amount of data generated over a 15-frame period. However, using the knowledge of the framing pattern, one can expect to achieve a faster detection by modifying the algorithm as follows. First fix the windowing parameter $a = 0.5$ and the memory size N = even multiple of p. We can treat frames from the three modes as three separate streams. Define $D_N^i(aN, n)$ as the signed Kolmogorov-Smirnov distance of the i^{th} stream, $i \in \{I, P, B\}$. Let $E_N(n)$ be the absolute value of the weighted average of the three distances. That is, $E_N(n) = |\frac{1}{15}D_N^I(aN, n) + \frac{4}{15}D_N^P(aN, n) + \frac{10}{15}D_N^B(aN, n)|$.

Fig. 2 illustrates the output of a broadcast-quality MPEG sequence, in Kb/frame. Note that the data rates of I, P, and B frames form three distinct groups. We apply the algorithm with memory size $N = 120$ and threshold $c = 0.75$. The circles at the top of the figure indicate the resulting detection points. They correspond well with visual partitions of the sequence. Our preliminary experiment using the scheme in Fig. 1 shows that changing the proportion of the two priorities at these points result in a smaller distortion than a fixed-priority policy.

5 Conclusion

In this paper we proposed a sequential change point detection algorithm that requires no assumption of the distributions before and after the change. We proved that it is asymptotically optimal as the memory size becomes large. The algorithm is general enough to be applicable in many contexts. It is presented here for the application of detecting changes in traffic models for variable-rate compression. By catering the compression parameters to each traffic model, the system can achieve better utilization of the resources.

Acknowledgements The authors are grateful to Profs. Peter Glynn and Vivek Borkar for their suggestions and Dr. Daniel Reininger for supplying the MPEG sequence.

References

[1] B. S. Darkovsky and B. E. Brodsky (1987) "A Nonparametric Method for Fastest Detection of a Change in the Mean of a Random Sequence", *Theory of Probability and its Applications*, 1987, pp. 640-648.

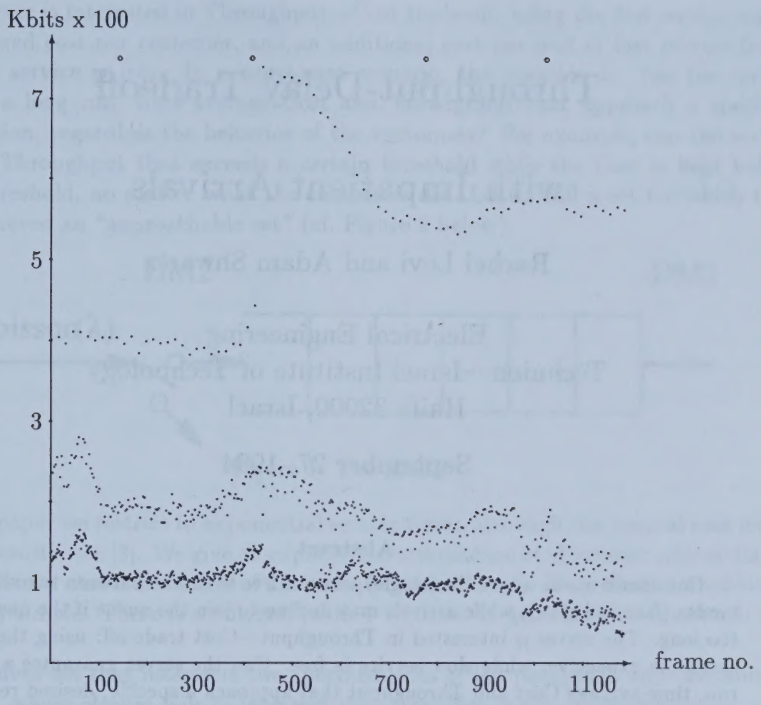


Figure 2: Output per frame of an MPEG sequence. The open circles represent the detection points.

[2] B. E. Brodsky and B. S. Darkhovsky (1993) *Nonparametric Methods in Change-Point Problems* Kluwer Academic Publishers.

[3] E. Carlstein (1988) "Nonparametric Change-Point Estimation", *Annals of Statistics*, 1988, vol. 16, no. 1, pp. 188-197.

[4] J. D. Gibbons and S. Chakraborti (1992) *Nonparametric Statistical Inference*, Marcel Dekker.

[5] H. Kroner, G. Hebuterne, P. Boyer, and A. Gravey (1991) "Priority Management in ATM Switching Nodes", *IEEE Journal on Selected Areas in Communications*, April 1991, vol. 9, no. 3, pp. 418-427.

[6] M. Pollak (1985) "Optimal Detection of a Change in Distribution", *Annals of Statistics*, 1985, pp. 206-227.

[7] D. Pollard (1984) *Convergence of Stochastic Processes*, Springer-Verlag.

[8] D. Tse, R. Gallager and J. Tsitsiklis (1993) "Optimal Buffer Control for Variable-Rate Lossy Compression", *31st Allerton Conference.*

Throughput-Delay Tradeoff with Impatient Arrivals

Rachel Levi and Adam Shwartz

Electrical Engineering
Technion—Israel Institute of Technology
Haifa 32000, Israel

September 27, 1994

Abstract

Consider a queue where the (single) server is free to choose between two service modes (fast and slow), while arrivals may decline to join the queue if the queue is too long. The server is interested in Throughput—Cost trade-off: using the fast service is expensive, while slow service is free. Can the server guarantee a long run, time average Cost and Throughput that approach a specific, desired region, regardless the behavior of the customers?

We characterize explicitly all convex approachable sets, as well as all non-approachable sets, as follows. There is a polygon so that:

Any convex set that intersects two sides of the polygon is approachable,

Any set whose closure fails to intersect these sides is not approachable.

The mathematical model that describes such systems is a Semi-Markov decision process with two independent decision makers, and vector payoff, where each coordinate is a long-run average. Under very mild conditions we obtain in [3] a general theory of approachability for such processes, which is applicable here.

1 Introduction

In modern communication networks, users not only compete for resources, but can also exert some control over the network (e.g. by deciding if and when to send large files, or submit heavy jobs.) Users also often have a choice between several networks, so that if the performance of one is not satisfactory, they can connect to another. Network management, on the other hand, may control the network to some extent in real time.

In this context, the classical building block of communication networks—the single server queue—gives rise to the following model. Consider a single server queue (M/•/1 type) with a Poisson arrival process and two allowed service rates (fast and slow.) The customers are impatient: they may decline to join the queue if the server is busy or if the queue is too long, although they always join if the queue is empty. Fig. 1 illustrates the structure of the queuing system.

The server is interested in Throughput—Cost trade-off: using the fast service mode entails a fixed cost per customer, and an additional cost per unit of fast service time, while slow service is free. In a worst-case scenario, the question is: can the server guarantee a long run, time average Cost and Throughput that approach a specific, desired region, regardless the behavior of the customers? For example, can the server guarantee Throughput that exceeds a certain threshold while the Cost is kept below another threshold, no matter what the customers do? Let us call a set for which this can be achieved an "approachable set" (cf. Figure 2 below).

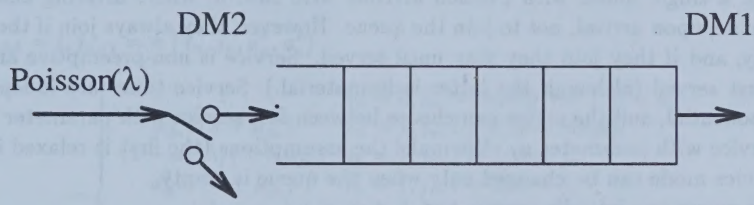


Figure 1: The Model

In this paper we restrict to exponential service times, although the general case leads to similar results: see [3]. We give an explicit characterization of all convex approachable sets, as well as a characterization of all non-approachable sets. This characterization is surprisingly simple. There is a polygon (with 4 vertices. In general there may be 5: see [3]) so that

Any convex set that intersects two specified sides of the polygon is approachable,

Any set whose closure fails to intersect the same sides is not approachable.

We provide closed-from (algebraic) expressions for the vertices (and hence a complete description of the polygons.) In addition, for each approachable set we provide an algorithm for computing a policy that guarantees that the long-time performance indeed approaches the desired set. In the case of exponential service we can compute a policy explicitly.

This model is a special case of a dynamic system with many decision makers and a vector of performance measures. Such systems appear in other engineering and communication systems such as many-user computers, and Data and Cellular communication networks. This approach to the analysis of performance regions of Markovian systems was proposed in Shimkin and Shwartz [2], where the discrete-time case was analyzed. Shimkin and Shwartz extended the idea of approachability due to Blackwell [1] from repeated games to stochastic games.

The theoretical foundation for the models we investigate is a "worst case scenario," where a single decision maker (called DM1) competes against all other decision makers, who have their own goals, which may contradicts those of DM1. DM1 is interested in a vector of time averaged performances, and he defines a required performance region. Can he guarantee a time averaged performance that reaches that desired set? What policy should he employ?

The mathematical model, introduced here to describe such systems is a Semi-Markov decision process with two independent decision makers, and with a vector payoff, where each coordinate is of the long-run average type. Under very mild conditions we obtain in [3] a general theory of approachability. This theory includes necessary and sufficient

conditions for a convex set to be approachable by DM1, sufficient conditions for any set to be approachable by DM1 as well as an "approaching" policy that achieves the desired performance when this is possible. This theory is the basis for the analysis of our queueing models, and includes the results of [1] and [2] as a special case.

2 The model

Consider a single queue with Poisson arrivals with rate λ, where arriving customers may decide, upon arrival, not to join the queue. However, they always join if the queue is empty, and if they join they stay until served. Service is non-preemptive and first come first served (although the latter is immaterial.) Service times are independent and exponential, and the server can choose between fast service with parameter μ_s and slow service with parameter μ_f. We make the assumptions (the first is relaxed in [3]):

- Service mode can be changed only when the queue is empty,
- The queue was initially empty, and $\lambda < \mu_s < \mu_f$.

We define a stage to be any instant when a customer arrives to the queue, or a service was completed. For each stage m, we define the corresponding state of the queue x_m to be the pair $x_m = (i_m, j_m)$, where $i_m \in \{0, 1, 2, \ldots\}$ denotes the number of customers in the system including the customer who is currently being served and $j_m \in \{empty, fast, slow\}$ denotes the service mode. (The system is at service mode *empty* when there are no customers in the system). We also denote by t_m the m-th transition time (i.e. from stage m to stage m+1), and by (a_m^1, a_m^2) the choices made by the server and the customer at stage m, respectively.

At each stage, we define the 'single stage throughput' and 'single stage cost' as follows: The single stage throughput Λ_m is the probability that the next decision instant will occur when a customer leaves the system, given the current state x_m, decisions (a_m^1, a_m^2) and transition time t_m. Equivalently, it is the probability a service will be completed at stage m+1 given (x_m, a_m^1, a_m^2, t_m). This probability does not dependent on the past behavior of either the server or the customers, nor on the initial state (see [3]). Thus we can write:

$$\Lambda_m = \Pr(\text{a service will be completed at stage m+1}). \tag{1}$$

The single stage cost C_m consists of a fixed cost, namely $c_1 \geq 0$, for each customer who left the system right after being served with the fast service, and $c_2 \geq 0$, for each unit time of using the fast service mode. Thus

$$C_m \equiv 1\{\text{a fast service was employed at stage m}\} \cdot (c_1 \Lambda_m + c_2 t_m) \tag{2}$$

where $1\{\}$ is the indicator function, i.e., $1\{A\} = 1$ iff the statement A is satisfied, and otherwise the value of the function is zero.

The performance criteria, i.e. Time-averaged-Throughput and Cost are:

$$\bar{\Lambda}_n = \frac{\sum_{m=0}^{n-1} \Lambda_m}{\sum_{m=0}^{n-1} t_m}, \quad \text{and} \quad \bar{C}_n = \frac{\sum_{m=0}^{n-1} C_m}{\sum_{m=0}^{n-1} t_m}. \tag{3}$$

Let $r_m = (C_m, \Lambda_m)$ be the vector payoff and $\bar{r}_m = (\bar{C}_m, \bar{\Lambda}_m)$ its time average.

The dynamic model is a Semi-Markov Decision Process with two independent Decision Makers. For simplicity we denote the server as DM1 (Decision Maker 1) and the customers as DM2 (Decision Maker 2). Define $a_n^1 \in \{fast, slow\}$ the service mode chosen by DM1 at stage n,
$a_n^2 \in \{0,1\}$ denotes DM2's action at stage n,
where $a_n^2 = 0$ means that he left the queue and $a_n^2 = 1$ means he joined it.
$\mu_{j_n} \in \{\mu_f, \mu_s\}$ is the service parameter at stage n.
The transition probabilities are:

$$p(i_{n+1} = l, j_{n+1} = k \mid i_n, j_n, a_n^1, a_n^2)$$

$$= \begin{cases} 1 & i_n = 0, j_n = empty, l = 1, k = a_n^1 \\ \mu_{j_n}/(\lambda + \mu_{j_n}) & i_n = 1, j_n, l = 0, k = empty \\ (1 - a_n^2)\lambda/(\lambda + \mu_{j_n}) & i_n = 1, j_n, l = 1, k = j_n \\ a_n^2\lambda/(\lambda + \mu_{j_n}) & i_n = 1, j_n, l = 2, k = j_n \\ \mu_{j_n}/(\lambda + \mu_{j_n}) & i_n > 1, j_n, l = i_n - 1, k = j_n \\ (1 - a_n^2)\lambda/(\lambda + \mu_{j_n}) & i_n > 1, j_n, l = i_n, k = j_n \\ a_n^2\lambda/(\lambda + \mu_{j_n}) & i_n > 1, j_n, l = i_n + 1, k = j_n \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

The transition times have the following distribution function (see [3]):

$$F_{x_n,x'}(t \mid a_n^1, a_n^2)$$

$$= \begin{cases} 1 - \exp(-\lambda t) & x_n = (0, empty), x' = (1, y) & y \in \{fast, slow\} \\ 1 - \exp(-(\lambda - \mu_f)t) & x_n = (l, fast), x' = (l \pm \{0,1\}, fast) & l = 1, 2, \ldots \\ 1 - \exp(-(\lambda - \mu_s)t) & x_n = (l, slow), x' = (l \pm \{0,1\}, slow) & l = 1, 2, \ldots \end{cases} \tag{5}$$

By (1)–(5), the Time-averaged-Throughput and Cost satisfy

$$\Lambda_n = \begin{cases} 0 & x_n = (0, empty) \\ \mu_f/(\lambda + \mu_f) & x_n = (l, fast), l = 1, 2, \ldots \\ \mu_s/(\lambda + \mu_s) & x_n = (l, slow), l = 1, 2, \ldots \end{cases} \tag{6}$$

$$C_n = \begin{cases} 0 & x_n = (0, empty) \\ (c_1\Lambda_n + c_2 t_n) & x_n = (l, fast), l = 1, 2, \ldots \\ 0 & x_n = (l, slow), l = 1, 2, \ldots \end{cases} \tag{7}$$

We let state 0 to be the state $0 := (0, empty)$, which will be proved below to satisfy a recurrence condition, and we let N to be the first passage stage to state 0, i.e. $N = \inf\{n > 0 : x_n = 0\}$.

To state the main result of this paper, we use the terms approachability and excludability: see definition 3.2 below. The results refer to Figure 2: explicit expressions for J, K, L and M are given in section 3.

Theorem 2.1. *(i) Let $B \subseteq \mathbb{R}^2$ be any closed convex set. Then B is approachable by the server* iff *it intersects both JM and KL. Otherwise is excludable by the arrivals. (ii) Any closed set $B' \subseteq \mathbb{R}^2$, which does not intersects either JM or KL is excludable by the arrivals.*

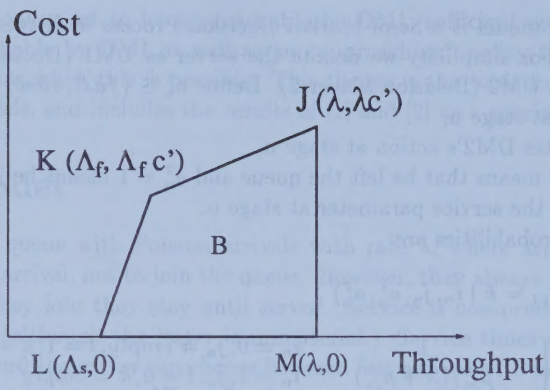


Figure 2: Performance Space

3 Approachability—definition and result

A policy for a decision maker can be any way of choosing an action based on the history of the system (i.e. its previous states and transition times) and the past actions (which includes a randomized decisions as well). We denote by π_i a policy for DMi (i=1,2), and by Π_i the class of all policies for DMi. A stationary policy is a policy that depends on the current state only. We denote by f (g) stationary policies for DM1 (DM2 respectively), and by $\Pi_1(S)$ ($\Pi_2(S)$) the class of all stationary policies.

For a given initial state $x_0 = x$, each pair of policies (π_1, π_2) induces a probability measure $P^x_{\pi_1,\pi_2}$. The corresponding expectation operator is denoted by $E^x_{\pi_1,\pi_2}$. A policy $\pi_1 \in \Pi_1$ is said to be stable if for some positive constants T_2 and R_2,

$$E^0_{\pi_1,\pi_2}\left(\sum_{m=0}^{N-1} t_m\right)^2 \leq T_2 \qquad \forall \pi_2 \in \Pi_2 \ , \tag{8}$$

$$E^0_{\pi_1,\pi_2}\left(\sum_{m=0}^{N-1} |r_m|\right)^2 \leq R_2 \qquad \forall \pi_2 \in \Pi_2 \ . \tag{9}$$

A set $\Pi_1' \subseteq \Pi_1$ is uniformly stable if (8) and (9) are satisfied for every $\pi_1 \in \Pi_1'$ with the same constants T_2 and R_2. Stability for DM2's policies is defined symmetrically.

Definition 3.1. *Let $\{y_n, n \geq 0\}$ be a sequence of random variables over a measurable space $(\Omega, \Im)$ and let $\{P_\lambda, \lambda \in \Lambda\}$ be a collection of probability measures on $(\Omega, \Im)$. We say that $y_n \overset{n\to\infty}{\longrightarrow} 0$ P_λ-a.s. at a uniform rate over Λ if, for all $\varepsilon > 0$:*

$$\lim_{K\to\infty} \sup_{\lambda\in\Lambda} P_\lambda\left(\sup_{n\geq K} |y_n| > \varepsilon\right) = 0 \ . \tag{10}$$

Let $d(r, B)$ denote the Euclidean point to set distance and $B^c_\delta = \{\beta \in \mathbb{R}^q : d(\beta, B) \geq \delta\}$.

Definition 3.2. *Let the initial state x be fixed. A set $B \subseteq \mathbb{R}^q$ is approachable (from x by DM1) if there exists a B-approaching policy $\pi_1^* \in \Pi_1$ such that $d(\bar{r}_n, B) \overset{n\to\infty}{\longrightarrow} 0$ for every $\pi_2 \in \Pi_2$, at a uniform rate over Π_2. B is excludable (from x by DM2) if there*

exists a B-excluding policy $\pi_2^* \in \Pi_2$ *such that for some* $\delta > 0$ $d(\bar{r}_n, B_\delta^c) \overset{n\to\infty}{\longrightarrow} 0$ *for every* $\pi_1^* \in \Pi_1$, *at a uniform rate over* Π_1.

Remarks:

1. Approachability and Excludability are obviously the same for a set and for its closure, thus we can restrict ourselves to closed sets.

2. Obviously, a given set can't be both approachable by DM1 and excludable by DM2. However, a set can be neither approachable nor excludable.

3. Suppose a closed set B is approachable by DM2. Then by definition for all $\delta > 0$, B_δ^c is excludable by DM2. The same holds for DM1, thus a sufficient (necessary) condition for approachability is also a sufficient (necessary) condition for excludability.

4. The requirement of the uniform rate of convergence is of great importance. From this requirement it follows, that given any approachable set and a related desired accuracy, DM1 is able to satisfy it, with finitely many stages.

Let

$$R(f,g) := \lim_{n\to\infty} \frac{E_{f,g}^0\left(\sum_{m=0}^{n-1} r_m\right)}{E_{f,g}^0\left(\sum_{m=0}^{n-1} t_m\right)} \tag{11}$$

where r_m is the vector payoff, and we use the notation:
$R(\Pi_1(S),g) := \{R(f,g) : f \in \Pi_1(S)\}$
$\bar{R}(\Pi_1(S),g) := convR(\Pi_1(S),g)$,
for the set of all possible ratios and its closed convex hull respectively. In a similar way we define $R(f,\Pi_2(S))$ and $\bar{R}(f,\Pi_2(S))$.

Theorem 3.3. *Assume the initial state is* 0 *and let* t *denote a generic transition time. Assume that the set of all policies are uniformly stable, and that there exist constants* a *and* b *such that for any pair of policies,* $a \leq E_{\pi_1,\pi_2}^0(t) \leq b$ *and the vector payoff is bounded. Let* $B \subseteq \mathbb{R}^q$ *be any closed convex set. Then (i)* B *is approachable by DM1* iff $B \cap \bar{R}(\Pi_1(S),g) \neq \emptyset$ *for any* $g \in \Pi_2(S)$. *(ii) If* B *is not approachable by DM1 then it is excludable by DM2, with a stationary policy.*

Proof: see [3].

Remark: For conditions under which the set of all policies are uniformly stable see [3] Lemma 2.1. Theorem 3.3 holds under weaker assumptions, see [3, Chapter 3].

4 Proofs

In this section we prove Theorem 2.1. We use the following notation.
$fast^\infty/slow^\infty$—The stationary deterministic policies of DM1.
g_1—The stationary deterministic policy when all the customers join the queue.
g_2—The stationary deterministic policy when the customers do not join the queue unless they must (i.e. unless the queue is empty).

Lemma 4.1. *Assume $\lambda < \mu_s < \mu_f$. Then Theorem 3.3 holds.*

Proof: The proof follows the following steps: we prove by coupling that $E^0_{\pi_1,\pi_2}(N^2)$ is uniformly bounded for any pair of policies. In the current queuing system the payoff vector is bounded and the transition times satisfy $a \leq E^0_{\pi_1,\pi_2}(t) \leq b$. Thus the set of all policies is uniformly stable and Theorem 3.3 holds. (see proof in [3, chapter 4]). Q.E.D.

To obtain an explicit expression we define the following constants (see Figure 2):

$$c' := (c_1 + c_2/\mu_f), \tag{12}$$

$$\Lambda_s := \frac{\lambda\mu_s}{\lambda + \mu_s}, \qquad \Lambda_f := \frac{\lambda\mu_f}{\lambda + \mu_f}. \tag{13}$$

Note that $\Lambda_s < \Lambda_f < \lambda$ and $c' > 0$. Let JM be the line segment between $J = (\lambda, \lambda c')$ and $M = (\lambda, 0)$, and let KL be the line segment between $K = (\Lambda_f, \Lambda_f c')$ and $L = (\Lambda_s, 0)$. Similarly we define JK and LM.

Proof of Theorem 2.1: The proof consists of the following three steps:
Step 1: Suppose that: (i) $JM = R(\Pi_1(S), g_1)$ and $KL = R(\Pi_1(S), g_2)$.
(ii) $JK = R(fast^\infty, \Pi_2(S))$ and $LM = R(slow^\infty, \Pi_2(S))$.
Let $B \subseteq \mathbb{R}^2$ be any closed convex set. Then B is approachable by DM1 *iff* it intersects both JM and KL, otherwise it is excludable by DM2. To see that, suppose the performance set B does not intersect JM and/or KL. Then DM2 has at least one stable policy, g_1 or g_2 respectively, such that $B \bigcap \bar{R}(\Pi_1(S), g_{1(2)}) = \emptyset$, and thus by Theorem 3.3 B is excludable by DM2. Now, suppose the performance set B does intersect JK and ML. B is a convex set and thus it intersects any line segment which connects any point in JM with any point in KL. From (ii) it then follows that for any $g \in \Pi_2(S)$, $B \bigcap \bar{R}(\Pi_1(S), g_1) \neq \emptyset$. Thus by Theorem 3.3, B is approachable by DM1. Thus, if we prove (i) and (ii) part (a) of Theorem 2.1 follows.
Step 2: Suppose (i) in step 1 holds. Then any closed set $B' \subseteq \mathbb{R}^2$, which does not intersect either JM or KL is excludable by DM2. To see that suppose $B' \bigcap JM \neq \emptyset$. By (i) $JM = \bar{R}(\Pi_1(S), g_1)$. Then by Theorem 3.3, JM is approachable by DM2. Now, since JM is a compact set and B' is closed set we have $d(B', \bar{R}(\Pi_1(S), g_1) = d(B', JM) > 0$. That means B' is excludable by DM2 (or in other words there exists a $\delta' > 0$, such that $B'^{\varepsilon}_{\delta'}$ is approachable by DM2). By following the same arguments we see that B' is excludable by DM2 if $B' \bigcap KL \neq \emptyset$. Thus, if we prove (i) part (b) of the Theorem follows.
Step 3: Here we prove (i) and (ii). Then by step 1 and step 2 the Theorem follows. To this end, note that [see [4] Th. 7.5]:

$$R(f,g) = \lim_{n\to\infty} \frac{E^0_{f,g}\left(\sum_{m=0}^{n-1} r_m\right)}{E^0_{f,g}\left(\sum_{m=0}^{n-1} t_m\right)} = \frac{E^0_{f,g}\left(\sum_{m=0}^{N-1} r_m\right)}{E^0_{f,g}\left(\sum_{m=0}^{N-1} t_m\right)}. \tag{14}$$

Denote $\bar{\Lambda}_N := \frac{E^0_{f,g}\left(\sum_{m=0}^{N-1} \Lambda_m\right)}{E^0_{f,g}\left(\sum_{m=0}^{N-1} t_m\right)}$ and $\bar{C}_N := \frac{E^0_{f,g}\left(\sum_{m=0}^{N-1} C_m\right)}{E^0_{f,g}\left(\sum_{m=0}^{N-1} t_m\right)}$.

Let f be any stationary policy for DM1. The policy f is well defined by the probability to choose the fast-service mode when the queue is empty. We denote this probability by α_0, i.e. $\alpha_0 = \Pr_f(a_n^1 = fast | x_n = (0, empty))$. Then by a smoothing argument, (6), (7) and (14) we have:

$$\bar{\Lambda}_N = \frac{\frac{\alpha_0\mu_f}{\mu_f+\lambda} E^0_{fast^\infty,g}(N-1) + \frac{(1-\alpha_0)\mu_s}{\mu_s+\lambda} E^0_{slow^\infty,g}(N-1)}{\frac{1}{\lambda} + \frac{\alpha_0}{\mu_f+\lambda} E^0_{fast^\infty,g}(N-1) + \frac{(1-\alpha_0)}{\mu_s+\lambda} E^0_{slow^\infty,g}(N-1)} \tag{15}$$

$$\bar{C}_N = \frac{\frac{c'\alpha_0\mu_f}{\mu_f+\lambda} E^0_{fast^\infty,g}(N-1)}{\frac{1}{\lambda} + \frac{\alpha_0}{\mu_f+\lambda} E^0_{fast^\infty,g}(N-1) + \frac{(1-\alpha_0)}{\mu_s+\lambda} E^0_{slow^\infty,g}(N-1)} \tag{16}$$

Proof of (i): Let $g = g_1$, and let f be any stationary policy for DM1. We proceed by calculating $E^0_{fast^\infty,g_1}(N)$ and $E^0_{slow^\infty,g_1}(N)$. We note that for both cases, N is twice the number of customers being served at the busy period of the system. Then by well known result for M/M/1 queue [[5] Chapter 5 Section 9]:

$$E^0_{fast^\infty,g_1}(N) = \frac{2\mu_f}{\mu_f - \lambda}, \quad \text{and} \quad E^0_{slow^\infty,g_1}(N) = \frac{2\mu_s}{\mu_s - \lambda}.$$

Substituting the above in (15) and (16) we obtain:

$$\bar{\Lambda}_N = \lambda, \quad \text{and} \quad \bar{C}_N = \frac{\lambda\alpha_0 c'\mu_f(\mu_s - \lambda)}{\mu_s(\mu_f - \lambda) + \lambda\alpha_0(\mu_s - \mu_f)}. \tag{17}$$

$\bar{C}_N$ in (16) is a continuous and increasing function of α_0 and thus

$$0 = \bar{C}_N(\alpha_0 = 0) \le \bar{C}_N(\alpha_0) \le \bar{C}_N(\alpha_0 = 1) = \lambda c',$$

which together with (14) implies $JM = \bar{R}(\Pi_1(S), g_1)$.

Now, let $g = g_2$. We calculate $E^0_{fast^\infty,g_1}(N)$ and $E^0_{slow^\infty,g_1}(N)$ directly. Suppose $f = fast^\infty$. Then the pair of policies $\{fast^\infty, g_2\}$ induce a Markov chain with only two states $\{(0, empty), (1, fast)\}$ and the transition matrix

$$\begin{pmatrix} 0 & 1 \\ \frac{\mu_f}{\lambda+\mu_f} & \frac{\lambda}{\lambda+\mu_f} \end{pmatrix}.$$

This Markov chain is irreducible and aperiodic and hence the required expectations are the inverse of the stationary probabilities [[6], Chapter 6, 2.19]. Thus $E^0_{fast^\infty,g_1}(N) = 2 + \frac{\lambda}{\mu_f}$ and using similar steps $E^0_{fast^\infty,g_1}(N) = 2 + \frac{\lambda}{\mu_f}$. Substituting in (15) and (16) we obtain

$$\bar{\Lambda}_N = \frac{\lambda\mu_s\mu_f}{\lambda\alpha_0\mu_s + \lambda(1-\alpha_0)\mu_f + \mu_f\mu_s} \tag{18}$$

$$\bar{C}_N = (\bar{\Lambda}_N - \Lambda_s)\frac{\mu_f c'(\lambda + \mu_s)}{\lambda(\mu_f - \mu_s)} \tag{19}$$

Note that $(\bar{C}_N, \bar{\Lambda}_N)$ in (18)–(19) satisfies the equation of the linear line through the points K and L. Moreover $\bar{C}_N$ is a continuous and increasing function of α_0 and thus

$$0 = \bar{C}_N(\alpha_0 = 0) \le \bar{C}_N(\alpha_0) \le \bar{C}_N(\alpha_0 = 1) = \lambda\Lambda_f c'$$

which, together with (13) implies $KL = \bar{R}(\Pi_1(S), g_2)$.

Proof of (ii): Suppose $f = fast^\infty$, and DM2 applies any stationary policy g. Then by (15) and (16):

$$\bar{C}_N = c'\bar{\Lambda}_N \tag{20}$$

$$\bar{\Lambda}_N = \frac{\frac{\mu_f}{\mu_f+\lambda} E^0_{fast^\infty,g}(N-1)}{\frac{1}{\lambda} + \frac{1}{\mu_f+\lambda} E^0_{fast^\infty,g}(N-1)} \tag{21}$$

Note that $(\bar{C}_N, \bar{\Lambda}_N)$ in (20)-(21) satisfies the equation of the linear line through the points J and K. $\bar{\Lambda}_N$ is a continuous and increasing function of α_0. By coupling arguments it can be shown that $E^0_{fast^\infty,g}(N)$ achieves its maximal value whenever $g = g_2$, and achieves its minimal value whenever $g = g_1$. Thus

$$\Lambda_f \leq \bar{\Lambda}_N(g = g_2) \leq \bar{\Lambda}_N(g) \leq \bar{\Lambda}_N(g = g_1) = \lambda.$$

Thus, by using (13), we have $JK = \bar{R}(fast^\infty, \Pi_2(S))$.

Following the same arguments for $f = slow^\infty$ we have $LM = \bar{R}(slow^\infty, \Pi_2(S))$, hence (ii) is satisfied. Q.E.D.

Note that the boundary of $JKLM$ is defined by the extreme policies $g_1, g_2, fast^\infty$ and $slow^\infty$. The line through the points J and K satisfies the equation $\bar{C} = c'\bar{\Lambda}$, which is obviously the case when the server always uses the fast service mode. Similarly, when the server always apply the slow service mode we have $\bar{C} = 0$. Also note that since the queue is stable (or the embedded Markov chain is positive recurrent for any pair of stationary policies) we have $\bar{\Lambda} = \lambda$ when all the customers join the queue (i.e. $g = g_1$). Remark: It can be shown by (13), (15) and (16) that for any fixed g, a fixed stationary policy of the customers, $\bar{C}_N$ is an affine function of $\bar{\Lambda}_N$ for any policy f of the server.

The approaching policy

The approaching policy is as follows: whenever at 0, let ρ be the current time averaged payoff. For any $\rho \notin B$, we define $C(\rho)$ to be a closest point in B to ρ and $H(\rho)$ to be the line through $C(\rho)$ that is perpendicular to $\rho - C(\rho)$. Then for any $\rho \notin B$:
(i) If $H(\rho)$ separates JK from ρ—use $fast^\infty$.
(ii) If $H(\rho)$ separates LM from ρ—use $slow^\infty$.
Otherwise—use $fast^\infty$ or $slow^\infty$. For $\rho \in B$ use any policy.

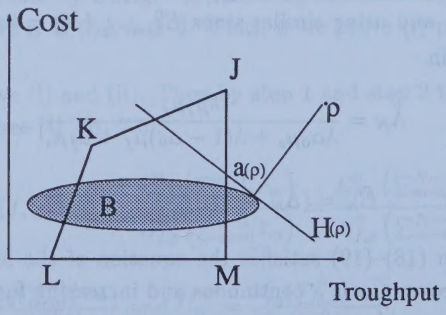


Figure 3: approaching policy

A performance set which is particularly interesting is a set with threshold values (Λ_0, C_0), so that the server (DM1), can guarantee a throughput of at least Λ_0 while the cost does not exceed C_0. We denote this set by B_0, $B_0 := \left\{(\bar{\Lambda}, \bar{C}) : \bar{\Lambda} \geq \Lambda_0, \bar{C} \leq C_0\right\}$. Then B_0 is approachable by the server *iff* it intersects JK. In other words the line JK is the set of all the optimal "thresholds sets," that the server can guarantee, in the sense that he cannot have better threshold sets with the two criteria simultaneously. JK represents the trade-off between the service cost and the throughput.

5 Extensions

Similar results were obtained for the case where the server can change the service mode at any stage. Under mild assumptions, we analyzed the approachable regions where the service times are arbitrary random variables: see [3]. These queuing systems are a special case of Semi-Markov decision process with two decision makers, for which we derive in [3] sufficient conditions for a set to be approachable, and necessary and sufficient conditions for a convex set to be approachable. The results in [1] and [2] are a special case of the results in [3].

Acknowledgments. Research of the second author was supported in part by the fund for promotion of research at the Technion.

References

[1] D. Blackwell, "An analog for the minmax theorem for vector payoffs", *Pacific J. Math.*, 6 pp. 1-9, 1956.

[2] N. Shimkin and A. Shwartz, "Guaranteed Performance Region in Markovian Systems with Competing Decision Makers", *IEEE Trans. Auto. Cont.*, Vol. 38, No. 1, Jan. 1993.

[3] R. Levi and A. Shwartz, *A Theory of Approachability and Throughput-Cost Tradeoff in a Queue with Impatient Customers*, EE Pub. 936, Technion, Haifa August 1994.

[4] S. M. Ross, *Applied Probability models with Optimization Application*, Holden-Day, San Francisco, 1970.

[5] L. Kleinrock, *Queuing systems*, Wiley-Interscience, New-York, 1975.

[6] E. Cinlar, *Introduction to stochastic processes*, Prentice-Hall, NJ., 1975.

STABILITY ANALYSIS OF HOPFIELD NEURAL NETWORKS WITH DELAYS*

Hui Ye Anthony N. Michel Kaining Wang
Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556, U.S.A.

Abstract

It is well known that Hopfield neural networks without delays exhibit no oscillations and possess global stability (i.e., all trajectories tend to some equilibrium). In the present paper we show that if the bound $\tau\beta\|T_2\| < 1$ is satisfied, then a corresponding Hopfield neural network with delays $\tau > 0$, interconnection matrix T_2 associated with delays, and gain of the neurons given by β, will exhibit similar qualitative properties as the original Hopfield neural network without delays ($\|T_2\|$ denotes the matrix norm induced by the Euclidean vector norm).

I. Introduction

Hopfield neural networks which constitute a class of artificial neural networks are of great common interest (see, e.g., [2]-[9]) and can be described by a system of differential equations of the form

$$\dot{x} = -Cx + TS(x) + b, \tag{H}$$

where $x \in R^n$ denotes the state variables associated with the neurons, b is a real n-vector representing bias terms, C is a real $n \times n$ diagonal matrix representing self-feedback terms, $T = T^T$ is a real symmetric $n \times n$ matrix representing neuron interconnections, and $S(x) = [s_1(x_1), \cdots, s_n(x_n)]^T$ is a real n-vector valued function whose components are sigmoidal nonlinearities representing the neurons. One of the reasons why Hopfield neural networks have received a great deal of attention is because these networks possess global stability (see, [4]), i.e., the outputs of the neurons will always converge to some equilibrium, and no oscillations will be present. Since Hopfield neural networks have the potential of performing parallel computation, some electronic implementations of Hopfield neural networks in VLSI technology have already been realized (see e.g., [10]-[12]).

*This work was supported in part by the National Science Foundation under Grant ECS91-07728.

However, in the implementation of artificial neural networks, time delays are unavoidably encountered, and it is known that time delays can cause systems to oscillate (see, e.g., [13]-[14], and [1]). Therefore, it is crucial to take time delays into consideration and to investigate the qualitative properties of Hopfield neural networks with delays. A class of such networks can be described by systems of equations of the form

$$\dot{x}(t) = -Cx(t) + T_1 S(x(t)) + T_2 S(x(t-\tau)) + b, \tag{D}$$

where x, b, C, S, T are the same as in (H), and $T_1 + T_2 = T$ (i.e., T_1 denotes the part of the interconnections associated with no delays while T_2 denotes the part of the interconnections associated with delays). We note that T_1, T_2 need not be symmetric; however we will assume that T is symmetric.

Hopfield neural networks with delays (D) have also received attention (see, e.g., [15]-[17]). Most of these works consider only local stability of system (D). Existing results concerning global stability are applicable to a special class of system (D) (for example, the components of $S(x)$ are piecewise linear functions instead of general sigmoidal functions (see, e.g., [16])). For the general Hopfield neural networks with delays given by equation (D), there are no existing results which provide explicit answers to the question whether system (D) is globally stable when the delay is sufficiently small. The above question was studied extensively in [1], and it is suggested in [1] that the answer to this question is affirmative.

In the present paper, we provide a bound for the delay τ of system (D) which depends on the interconnection matrix and on the neuron gains. We prove rigorously that when the delay is smaller than this bound, system (D) will have the same global stability and also the same local stability properties as system (H). As a consequence of this, it can be concluded that the system (D) will be globally stable for sufficiently small delays when the corresponding system (H) is globally stable. In our proof, we make use of an energy functional for system (D) and we show that this energy functional decreases along the solutions of (D), ultimately converging to some equilibrium of system (D). We also show that any (asymptotically) stable equilibrium of (D) corresponds to a local minimum of the energy functional. When delays are smaller than the bound mentioned above, we prove that the set of all (asymptotically) stable equilibria of (D) is identical to the set of all the (asymptotically) stable equilibria of (H). In other words, not only the global stability of system (D), but also the local (asymptotic) stability of each equilibrium of (D) will be unaffected by small delays. Moreover, we establish an effective criterion for the (asymptotic) stability of each single equilibrium of (D).

II. Notation

Let R denote the set of real numbers and let R^n denote real n-space. If $x \in R^n$, then $x^T = (x_1, \cdots, x_n)$ denotes the transpose of x. Let $R^{n \times m}$ denote the set of $n \times m$ real

matrices. If $B = [b_{ij}]_{n\times m} \in R^{n\times m}$, then B^T denotes the transpose of B, and $det(B)$ denotes the determinant of B. For $x \in R^n$, let $\|x\|$ denote the Euclidean vector norm, $\|x\| = (x^T x)^{1/2}$, and for $A \in R^{n\times n}$, let $\|A\|$ denote the norm of A induced by the Euclidean vector norm, i.e., $\|A\| = [\lambda_{max}(A^T A)]^{\frac{1}{2}}$. I denotes the identity $n \times n$ matrix.

Let R^+ denote the set of nonnegative real numbers, i.e., $R^+ = [0, +\infty)$. Let X be a subset of R^n and let Y be a subset of R^m. We denote by $C[X, Y]$ the set of all continuous functions from X to Y, and we denote by $C^k[X, Y]$ the set of all functions from X to Y which have continuous derivatives up to order k. Let $\tau > 0$, $x \in C[[-\tau, +\infty), R^n]$, and $t > 0$. We define $x_t \in C[[-\tau, 0], R^n]$ as $x_t(s) = x(t + s)$ for $s \in [-\tau, 0]$. For any $\phi \in C[[-\tau, 0], R^n]$, the norm of ϕ, denoted by $|\phi|$, is defined as $|\phi| = max\{\|\phi(t)\| : t \in [-\tau, 0]\}$.

The system (D) is said to be globally stable if for any solution $x(t)$, $\lim_{t\to\infty} x(t)$ exists. The definitions of stability and asymptotic stability of any equilibrium of (D) are contained in standard texts (see, e.g., [18]).

III. Global Stability of Hopfield Neural Networks with Delays

In the present section, we consider Hopfield neural networks with delays described by retarded type differential-difference equation of the form

$$\dot{x}(t) = -Cx(t) + T_1 S(x(t)) + T_2 S(x(t-\tau)) + b, \tag{D}$$

where $C = diag\{c_1, \cdots, c_n\}$ with $c_i > 0$ for $i = 1, \cdots, n$, $T = T_1 + T_2$ is a symmetric matrix, $b \in R^n$, $\tau > 0$, and $S(x) = [s_1(x_1), \cdots, s_n(x_n)]^T$ is a sigmoidal vector function such that $s_i(\cdot) \in C^1(R, R)$, $s_i'(\rho) \triangleq \frac{ds_i}{d\rho}(\rho) > 0$, $\lim_{\rho\to\infty} s_i(\rho) = 1$, $\lim_{\rho\to-\infty} s_i(\rho) = -1$, and $\lim_{|\rho|\to\infty} s_i'(\rho) = 0$ for $i = 1, \cdots, n$. In order to establish our main results, we need to present some existing properties for system (D).

Lemma 1 Any solution of system (D) is bounded.

Remark 1 The proof of Lemma 1 is identical to the proof when $\tau = 0$ (see, e.g., [3]), noticing that the c_i's are positive and the $s_i(x_i)$'s are bounded, $i = 1, \cdots, n$.

Assumption A For any equilibrium x_e of system (D) (i.e., $-Cx_e + TS(x_e) + b = 0$ with $T = T_1 + T_2$), the matrix $J(x_e)$ is a nonsingular matrix, where

$$J(x) = -T + diag\{c_i(s_1'(x_1))^{-1}, \cdots, c_n(s_n'(x_n))^{-1}\}. \tag{J}$$

Lemma 2 For almost all $b \in R^n$ (except a set with Lebesgue measure 0), system (D) satisfies Assumption A.

Remark 2 Lemma 2 can easily be proved by using Sard's Theorem (see Lemma 3.3 of [7]).

Lemma 3 When system (D) satisfies Assumption A, the set of equilibria of system (D) is a discrete set.

Remark 3 Lemma 3 can be proved by the Inverse Function Theorem. (see Remark 3.4 of [7]).

Remark 4 Throughout this paper, we will assume that system (D) satisfies Assumption A. We note that this assumption is also required in global stability studies of artificial neural networks without delays (see, e.g., [7], [3]). In fact, Assumption A is very mild, since by Lemma 2 this assumption is satisfied for almost all $b \in R^n$.

We are now in a position to establish the following result.

Theorem 1 Suppose that Assumption A is satisfied for system (D), and suppose that

$$\tau\beta\|T_2\| < 1, \tag{3.1}$$

where $\beta = \max_{x\in R^n} \|D(x)\|$, and $D(x) \triangleq diag\{s_1'(x_1), \cdots, s_n'(x_n)\}$. Then, system (D) is globally stable.

Proof: Let $y = S(x)$, and suppose x_t is a function in $C([-\tau, 0], R^n)$. Then $y_t = S(x_t)$ is also in $C([-\tau, 0], R^n)$. We define an energy functional $E(x_t)$ associated with (D) by

$$\begin{aligned} E(x_t) &= -y_t^T(0)Ty_t(0) + 2\sum_{i=1}^{n} \int_0^{(y_t(0))_i} c_i s_i^{-1}(\sigma)d\sigma - 2y_t^T(0)b \\ &\quad + \int_{-\tau}^{0} [y_t(\theta) - y_t(0)]^T T_2^T f(\theta) T_2 [y_t(\theta) - y_t(0)]d\theta, \end{aligned} \tag{3.2}$$

where $f(\theta) \in C^1([-\tau, 0], R^+)$ which will be specified later. After changing integration variables, (3.2) can be represented by

$$\begin{aligned} E(x_t) &= -y^T(t)Ty(t) + 2\sum_{i=1}^{n} \int_0^{y_i(t)} c_i s_i^{-1}(\sigma)d\sigma - 2y^T(t)b \\ &\quad + \int_{t-\tau}^{t} [y(w) - y(t)]^T T_2^T f(w-t) T_2 [y(w) - y(t)]dw. \end{aligned} \tag{3.3}$$

The derivative of $E(x_t)$ with respect to t along any solution of (D) can be computed as

$$\frac{dE(x_t)}{dt} = -\int_{-\tau}^{0} \alpha(x_t, \theta)^T M(x_t, \theta)\alpha(x_t, \theta)d\theta, \tag{3.4}$$

(involving several steps which have been omitted), where $f'(\theta) = \frac{df}{d\theta}(\theta)$ and $\alpha(x_t, \theta)^T = [\alpha_1^T, \alpha_2^T, \alpha_3^T]^T$ such that

$$\alpha_1 = -Cx(t) + T_1 y(t) + T_2 y(t-\tau) + b, \tag{3.5}$$

$$\alpha_2 = T_2[y(t-\tau) - y(t)], \tag{3.6}$$

$$\alpha_3 = T_2[y(t+\theta) - y(t)], \tag{3.7}$$

and

$$M(x_t, \theta) = \begin{bmatrix} \frac{2D(x(t))}{\tau} & -\frac{D(x(t))}{\tau} & D(x(t))T_2^T f(\theta) \\ -\frac{D(x(t))}{\tau} & \frac{f(-\tau)}{\tau} I & 0 \\ f(\theta)T_2 D(x(t)) & 0 & f'(\theta)I \end{bmatrix}, \tag{3.8}$$

where I denotes the $n \times n$ indentity matrix. We will show that if the hypotheses of Theorem 1 are satisfied, then $M(x_t, \theta)$ is positive definite for all $\theta \in [-\tau, 0]$ and all x_t which satisfy equation (D). In doing so, we let $U = U_3 U_2 U_1$, where

$$U_1 = \begin{bmatrix} I & 0 & 0 \\ -\frac{I}{2} & I & 0 \\ 0 & 0 & I \end{bmatrix}, U_2 = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ -\frac{\tau}{2} f(\theta) T_2 & 0 & I \end{bmatrix},$$

and

$$U_3 = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & -\frac{1}{2} f(\theta) T_2 D(x(t)) [\frac{f(-\tau)}{\tau} I - \frac{D(x(t))}{2\tau}]^{-1} & I \end{bmatrix}.$$

It is not difficult to verify that $\tilde{M} = U M(x_t, \theta) U^T$ is a diagonal matrix. In fact,

$$\tilde{M} = diag\{M_1, M_2, M_3\} \tag{3.9}$$

where

$$M_1 = \frac{2D(x(t))}{\tau}, \tag{3.10}$$

$$M_2 = \frac{f(-\tau)}{\tau} I - \frac{D(x(t))}{2\tau}, \tag{3.11}$$

and

$$M_3 = f'(\theta) I - \frac{f(\theta) T_2 D(x(t))}{2} [(\frac{f(-\tau)}{\tau} I - \frac{D(x(t))}{2\tau})^{-1} + 2\tau D^{-1}(x(t))] \frac{D(x(t)) T_2^T f(\theta)}{2}. \tag{3.12}$$

It now follows that $M(x_t, \theta)$ is positive definite if and only if $\tilde{M}$ is positive definite, and if and only if M_1, M_2 and M_3 are all positive definite.

If $\tau\beta\|T_2\| < 1$ and, if we chose f as

$$f(\theta) = l\Big[\beta\tau^2\|T_2\|^2 - \frac{(\tau + \theta)\|T_2\|^2\beta\tau}{2 - \beta^2\tau^2\|T_2\|^2}\Big]^{-1}, \tag{3.13}$$

where l can be any number such that $1 > l > (2 - \beta^2\tau^2\|T_2\|^2)^{-1}$, then it can be shown that M_1, M_2, and M_3 are positive definite for all x_t which satisfy equation (D) and for all θ. (Due to space limitations, we omit the details for this derivation.) Furthermore, we know that with the choice of f given by (3.13), $M(x_t, \theta)$ is positive definite and therefore, $\frac{dE(x_t)}{dt} \leq 0$ along any solution x_t of (D).

Since for any x_t satisfying equation (D), x_t is bounded (see Lemma 1), and since $\frac{dE(x_t)}{dt} \leq 0$, it follows from the Invariance Theory (see ch. 4, Lemmas 1.4 and 2.1 of [18]), that the limit set of x_t as $t \to \infty$ is a connected subset of the set of all equilibria of system (D). By Assumption A, this set is a discrete set (Lemma 3). Thus we have proved that x_t approaches some equilibrium of system (D) as $t \to \infty$. Specifically, we have shown that $\lim_{t\to\infty} x(t)$ exists. □

IV. Local Stability of Hopfield Neural Networks with Delays

In the preceding section we showed that when $\tau\beta\|T_2\| < 1$, Hopfield neural networks with delays described by equation (D) are globally stable, i.e., any solution of (D) will converge to *some* equilibrium of (D). Since in the implementation of Hopfield neural networks as associative memories, information is stored in specific asymptotically stable equilibria (called stable memories), good criteria which ensure the asymptotic stability of an equilibrium of (D) are of great interest. We address this issue in the present section.

Definition An element $\phi \in C([-\tau, 0], R^n)$ is called a *local minimum* of the energy functional E defined by (3.2) if there exists a $\delta > 0$, such that for any $\tilde{\phi} \in C([-\tau, 0], R^n)$, $E(\phi) \leq E(\tilde{\phi})$ whenever $|\phi - \tilde{\phi}| < \delta$.

We are now able to establish the following results.

Theorem 2 Suppose that the conditions of Theorem 1 are satisfied. If x_e is an equilibrium of (D), then the following statements are equivalent:

(1) x_e is a stable equilibrium of (D);

(2) x_e is an asymptotically stable equilibrium of (D);

(3) ϕ_{x_e} is a local minimum of the energy functional E given by (3.2), where $\phi_{x_e} \in C([-\tau, 0], R^n)$ such that $\phi_{x_e} \equiv x_e$;

(4) $J(x_e)$ is positive definite, where $J(x)$ is given in equation (J).

Proof:

a) (1) $\Rightarrow$ (2): Since Assumption A is satisfied, the set of equilibria of system (D) is a discrete set by Lemma 3. Therefore, when $\epsilon > 0$ is sufficiently small, there is no other equilibrium in $U(x_e, \epsilon)$, a neighborhood of x_e, given by

$$U(x_e, \epsilon) \triangleq \{x \in R^n : \|x - x_e\| < \epsilon\}. \tag{4.1}$$

Since x_e is a stable equilibrium of (D), there exists an $\eta > 0$ such that for any $\phi \in C([-\tau, 0], R^n)$ satisfying $|\phi - x_e| < \eta$, $|x_t - x_e| < \epsilon$ for all $t > 0$ where x_t is the solution of (D) with initial condition ϕ. Thus $x_t \in C([-\tau, 0], U(x_e, \epsilon))$ for all t. In view of Theorem 1, x_t will converge to some equilibrium of system (D). Since x_e is the only equilibrium of (D) in $U(x_e, \epsilon)$, it follows that x_t converges to x_e. Thus, we have shown that x_e is an attractive equilibrium of system (D). Therefore the stable equilibrium x_e of (D) is an asymptotically stable equilibrium of system (D).

b) (2) $\Rightarrow$ (3): Since x_e is an asymptotically stable equilibrium of system (D), there exists an $\eta > 0$ such that for any $\phi \in C([-\tau, 0], R^n)$ satisfying $|\phi - x_e| < \eta$, x_t converges to x_e where x_t is the solution of (D) with initial condition ϕ. Therefore $E(\phi_{x_e}) \leq E(x_t) \leq E(\phi)$ for any $\phi \in C([-\tau, 0], R^n)$ satisfying $|\phi - x_e| < \eta$. Therefore ϕ_{x_e} is a local minimum of the energy functional E.

c) (3) $\Rightarrow$ (4): Let $\tilde{E}$ be a function from R^n to R defined by

$$\tilde{E}(x) \triangleq -y^T T y + 2\sum_{i=1}^{n} \int_0^{y_i} c_i s_i^{-1}(\sigma) d\sigma - 2y^T b, \tag{4.2}$$

where $y = S(x)$. Comparing E with $\tilde{E}$, we note that $\tilde{E}$ is a function defined on R^n while E is a functional defined on $C([-\tau, 0], R^n)$. Since ϕ_{x_e} is a local minimum of E, x_e must be a local minimum of $\tilde{E}$. For otherwise, there would exist a sequence $\{x_n\} \subset R^n$ such that $x_n \to x_e$ as $n \to \infty$ and $\tilde{E}(x_n) < \tilde{E}(x_e)$. Let ϕ_{x_n} denote the constant function $\phi_{x_n} \equiv x_n$ in $C([-\tau, 0], R^n)$. Then $|\phi_{x_n} - \phi_{x_e}| \to 0$ as $n \to \infty$, and

$$E(\phi_{x_n}) = \tilde{E}(x_n) < \tilde{E}(x_e) = E(\phi_{x_e}).$$

This contradicts the fact the ϕ_{x_e} is a local minimum of E. Therefore, x_e is a local minimum of $\tilde{E}$. Hence, $\tilde{J}(x_e)$ is positive semidefinite (see, e.g., Theorem 3.6 of [24]), where $\tilde{J}(x)$ is the Hessian matrix of $\tilde{E}$ given by

$$\tilde{J}(x) = \left[\frac{\partial^2 \tilde{E}}{\partial x_i \partial x_j}\right]_{n \times n}. \tag{4.3}$$

It can be shown that

$$\tilde{J}(x) = 2D(x)J(x)D(x),$$

where $D(x) = diag\{s_1'(x_1), \cdots, s_n'(x_n)\}$, and $J(x)$ is given by equation (J). Therefore, $J(x_e)$ is also positive semidefinite. By Assumption A, $J(x_e)$ is a nonsingular matrix. Thus we have shown that $J(x_e)$ is positive definite.

d) (4) $\Rightarrow$ (1): We need to prove that x_e is a stable equilibrium of system (D), i.e., for any $\epsilon > 0$, there exists a $\delta > 0$ such that for any $\phi \in C([-\tau, 0], R^n)$, if $|\phi - x_e| < \delta$, then $|x_t - x_e| < \epsilon$ where x_t is the solution of (D) with initial condition ϕ.

Since $J(x_e)$ is positive definite, then $\tilde{J}(x_e)$ must also be positive definite where $\tilde{J}(x)$ is the Hessian matrix of $\tilde{E}$ given by (4.4). Furthermore,

$$\nabla_x \tilde{E}(x) = 2(-Ty + Cx - b)D(x)$$

where $D(x)$ is given in part (b). Therefore, $\nabla_x \tilde{E}(x_e) = 0$ since x_e is an equilibrium of (D). It follows (by Theorem 3.6 of [24]) that x_e is a local minimum of $\tilde{E}$, i.e., there exists a $\delta_1 > 0$, $\delta_1 < \epsilon$, such that whenever $0 < \|x - x_e\| \leq \delta_1$, $\tilde{E}(x_e) < \tilde{E}(x)$. Let $r = \min\{\tilde{E}(x) : \|x - x_e\| = \delta_1\}$. Then it is true that $r > \tilde{E}(x_e)$. Since $E(\phi_{x_e}) = \tilde{E}(x_e)$, it follows that $r > E(\phi_{x_e})$. Note that E is a continuous functional. Therefore, there exists a $\delta > 0$ such that whenever $|\phi - x_e| < \delta$, where $\phi \in C([-\tau, 0], R^n)$, we have $E(\phi) < r$. Suppose x_t is any solution of (D) with the initial condition ϕ such that $|\phi - x_e| < \delta$. We will show that $|x_t - x_e| < \delta_1 < \epsilon$. For otherwise, there would exist a $t_0 > 0$ such that $\|x_{t_0}(0) - x_e\| = \delta_1$, i.e., $\|x(t_0) - x_e\| = \delta_1$. By the definition of E and $\tilde{E}$, we have $E(x_{t_0}) \geq \tilde{E}(x(t_0)) \geq r$. Therefore, we obtain $E(x_{t_0}) > E(\phi)$ which contradicts the fact that E is monotonically decreasing along any solution of (D). Thus, we have shown that x_e is an asymptotically stable equilibrium of system (D). □

Remark 5 We note that statement (4) in Theorem 2 is independent of the delay τ. Therefore, if system (D) satisfies Assumption A, and if the condition $\tau\beta\|T_2\| < 1$ is satisfied, then the locations of the (asymptotically) stable equilibria of system (D) will

not depend on the delay τ. This is true if in particular $\tau = 0$. Therefore, if $\tau\beta\|T_2\| < 1$, then system (D) and system (H) (obtained by letting $\tau = 0$ in (D)) will have identical (asymptotically) stable equilibria. We state this in the form of a corollary.

Corollary 1 Under the conditions of Theorem 1, x_e is an (asymptotically) stable equilibrium of system (D) if and only if x_e is an (asymptotically) stable equilibrium of system (H). This is true if and only if $J(x_e)$ is positive definite, where $J(x)$ is given in equation (J). □

Remark 6 Corollary 1 provides an effective criterion for testing the (asymptotic) stability of any equilibrium of Hopfield neural networks with delays described by (D). This criterion constitutes necessary and sufficient conditions, as long as $\tau\beta\|T_2\| < 1$. □

V. An Example

To illustrate the applicability of some of the preceding results, we consider the system

$$\dot{x}(t) = -Cx(t) + TS(x(t-\tau)), \tag{5.1}$$

where $x \in R^2$, $C = diag\{1.1, 1.2\}$, and $S(x) = [s_1(x_1), s_2(x_2)]$ such that

$$s_1(x_1) = \frac{2}{\pi} tan^{-1}\Big(\frac{1.4\pi}{2} x_1\Big) \quad \text{and} \quad s_2(x_2) = \frac{2}{\pi} tan^{-1}\Big(\frac{1.5\pi}{2} x_2\Big),$$

where T is a symmetric matrix given by

$$T = \begin{bmatrix} -0.2 & 1 \\ 1 & -0.1 \end{bmatrix}.$$

System (5.1) has three equilibria given by $x_{e_0} = (0,0), x_{e_1} = (0.2140, 0.2091)$ and $x_{e_2} = (-0.2140, -0.2091)$. The classical method of analyzing the stability of (5.1) is to linearize (5.1) about each of its equilibria. For example, the linearization of (5.1) at the equilibrium $x_{e_1} = (0.2140, 0.2091)$ is given by

$$\dot{y}(t) = -Cy(t) + TS_0 y(t-\tau), \tag{5.2}$$

where $y = x - x_{e_1}$, C and T are the same as in (5.1), and

$$S_0 = \begin{bmatrix} 1.1462 & 0 \\ 0 & 1.2071 \end{bmatrix}.$$

The results in [1] cannot be applied in the present case, since system (5.2) cannot be decomposed into two one dimensional subsystems which is essential to the derivation of the results in [1].

Although, there exist in the literature some sufficient conditions for the asymptotic stability of linear time-delay systems given by

$$\dot{x}(t) = Ax(t) + Bx(t-\tau), \tag{5.3}$$

where A, B are constant $n \times n$ matrices, these results are in general very restrictive when applied to system (5.2). For example, the results of [23] yield

$$2\tau(\|A\| + \|B\|)\left(\frac{\lambda_{max}(P)}{\lambda_{min}(P)}\right)^{\frac{1}{2}}\|PB\| < 1, \tag{5.4}$$

as a condition for the asymptotic stability of the equilibrium $x = 0$ of system (5.3), where P is the positive definite matrix such that $(A + B)^T P + P(A + B) = -I$, and I is the identity matrix. When the bound given in (5.4) is applied to system (5.2), we obtain that the equilibrium $y = 0$ (or $x = x_{e_1}$) is asymptotically stable if the delay $\tau < 0.014$. Thus, by the classical method of linearization, we know that the equilibrium x_{e_1} of system (5.1) is asymptotically stable if $\tau < 0.014$. If we apply Corollary 1 to system (5.1), it is easily shown that $J(x_{e_1})$ and $J(x_{e_2})$ are positive definite, and $J(x_{e_0})$ is not positive definite, where $J(x)$ is defined by equation (J). Thus, we know by Corollary 1 that when $\tau < 0.579$, x_{e_1} and x_{e_2} are (asymptotically) stable equilibria of system (5.1) while x_{e_0} is not (asymptotically) stable. Therefore, for the present example, Corollary 1 provides stability conditions which are significantly less restrictive than existing results. Additionally, Theorem 1 shows that that system (5.1) is globally stable when $\tau < 0.579$.

References

[1] C. M. Marcus and R. M. Westervelt, "Stability of analog neural networks with delay", *Physical Review A*, 39, pp. 347-359, 1989.

[2] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities", *Proc. Natl. Acad. Sci. U.S.A.*, vol. 79, pp. 2554-2558, 1982.

[3] M. Cohen and S. Grossberg, "Absolute stability of global pattern formation and parallel memory storage by competitive neural networks", *IEEE Trans. Syst. Man., Cybern.*, vol. SMC-13, pp. 815-826, 1983.

[4] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons", *Proc. Natl. Acad. Sci. U.S.A.*, vol. 81, pp 3088-3092, 1984.

[5] L. Personnaz, I. Guyon, and G. Dreyfus, "Collective computational properties of neural networks: New learning mechanisms", *Physical Review A*, vol. 34, pp. 4217-4228, 1986.

[6] , L. Personnaz, I. Guyon, and G. Dreyfus, "Information storage and retrieval in spin-glass like neural networks", *J. of Phys. Letter*, vol. 46, pp. L359-L365, 1985.

[7] J. H. Li, A. N. Michel, and W. Porod, "Qualitative analysis and synthesis of a class of neural networks", *IEEE Trans. Circuits Syst.*, vol. CAS-35, no. 8, pp. 976-986, 1988.

[8] A. N. Michel, J. A. Farrell, and W. Porod, "Qualitative theory of neural networks", *IEEE Trans. Circuits Syst.*, vol. CAS-36, no. 2, pp. 229-243, 1989.

[9] L. O. Chua and L. Yang, "Cellular neural networks: Theory", *IEEE Trans. Circuits Syst.*, vol. 35, pp. 1273-1290, 1988.

[10] J. J. Hopfield and D. Tank, "Computing with neural circuits: A model", *Science*, vol. 2, pp. 625-633, 1986.

[11] D. Tank and J. J. Hopfield, "Collective Computation in Neuronlike Circuits", *Scientific American*, 257, 6, pp. 104-114.

[12] J. S. Denker, " Neural networks for computing ", *Amer. Inst. Phys. Conf. 151*, Snowbird, Utah, 1986.

[13] B. D. Coleman and G. H. Renninger, "Periodic Solutions of Certain Nonlinear Integral Equations with a Time Lag", , *SIAM J. App. Math.*, vol 31, pp. 111, 1976.

[14] U. an der Heiden, *Analysis of neural networks*, Springer, New York, 1980.

[15] T. A. Burton, "Averaged neural networks", *Neural Networks*, vol. 6, pp. 677-680, 1993.

[16] P. P. Civalleri, M. Gilli, and L. Pandolfi, "On stability of cellular neural networks with delay", *IEEE Trans. Circuits Syst.*, vol. 40, pp. 157-165, 1993.

[17] H. Ye, A. N. Michel, and K. Wang, "Robust stability of nonlinear time-delay systems with applications to neural networks", *Submitted to IEEE Trans. Circuits Syst..*

[18] J. Hale, *Theory of Functional Differential Equations, Springer-Verlag, 1977 .*

[19] A. Avez, *Differential Calculus*, chap. 6, New York: Wiley, 1986.

[20] E. Cheres, Z. L. Palmor, and S. Gutman, "Quantitative measures of robustness for systems including delayed perturbations" *IEEE Trans. Auto. Control, Vol. 34, pp. 1203-1204, 1989.*

[21] H. Trinh and M. Aldeen, "On the stability of linear systems with delayed perturbations" *IEEE Trans. Auto. Control, to appear.*

[22] Li-Ming Li, "Stability of linear neutral delay-differential systems" *Bull. Austral. Math. Soc. 38, 1988, pp 339-344.*

[23] D. Ya. Khusainov and E. A. Yun'kova, "Estimation of magnitude of retardation in linear differential systems with deviated argument", *Ukrainskii Matematicheskii Ahurnal*, vol. 35, no. 2, pp. 261-264, 1981

[24] C. Goffman, *Calculus of Several Variables*, New York, Harper & Row, 1965.

Supervised Learning in Neural Networks without Explicit Error Back-Propagation

Robert D. Brandt[1] and Feng Lin[2]

[1] Intelligent Devices, Inc., 465 Whittier Avenue, Glen Ellyn, IL 60137
[2] Department of ECE, Wayne State University, Detroit, MI 48202

Abstract

We describe a model for the implementation of supervised learning algorithms for networks of sigmoidal "neurons" which does not require that error feedback be explicitly provided by means of a dedicated feedback network. Instead, this model makes use of a locally-defined environmental gradient which is implicit in the strengths of synapses, their rates of change, and pre- and post-synaptic activity levels. One of the algorithms suggested by this model is numerically isomorphic to the two-phase (feed-forward and feed-backward) error back-propagation algorithm and can be expected to behave identically iteration-to-iteration to that algorithm. However, other formulations suggested by this model allow a phaseless and even asynchronous and concurrent processing by the neurons. This simplifies digital simulation and hardware implementation of neural networks and potentially allows training to be implemented in analog hardware. Furthermore, the elimination of strict requirements for a highly structured and tightly coordinated feedback network and distinct feed-forward and feed-backward phases directly addresses important issues raised by neuroscientists about the plausibility of similar adaptation occurring in biological neural systems.

1 Introduction

We propose a different model for the implementation of supervised learning algorithms for networks of sigmoidal "neurons" which does not require that error feedback be explicitly provided by means of a dedicated feedback network. In this model, a locally-defined *environmental gradient* which is implicit in the strengths of synapses, their rates of change, and pre- and post-synaptic activity levels is used in the adaptation. This environmental gradient always exists and is generally non-zero, independently of the presence of a supervisor, so long as there is some change in synaptic strength, regardless of the driving force behind the modification, much as a *Hebbian gradient* always exists at any synapse. Because no collateral feedback network need be simulated, the connectivity of trainable networks are exactly the same as corresponding non-trainable networks (i.e. networks with fixed synapse strength). Trainable artificial neural network hardware designed on this model would be able to more easily incorporate a capacity for dynamic reconfigurability, because there is no feedback network to be reconfigured.

One of the algorithms suggested by this model is numerically isomorphic to the two-phase (feed-forward and feed-backward) error back-propagation algorithm and will behave identically iteration-to-iteration to that algorithm. However, the formulations suggested by this model allow more than just simplified simulation and hardware. They also allow a phaseless and even asynchronous and concurrent processing by the neurons. In addition, it potentially allows training to be implemented in analog hardware. In an experiment not reported here, a solution of the exclusive-or problem was learned in a numerical simulation of an analog network (a continuous-time dynamical system).

More importantly, the elimination of strict requirements for a highly structured and tightly coordinated feedback network and distinct feed-forward and feed-backward phases directly addresses important issues of biological plausibility. This will be discussed in the next section. The subsequent section presents the main results.

2 Neurobiological Issues

In this section, we briefly mention some issues (questions) raised by some of our colleagues in neuroscience regarding the relevance of this type of network model to neuroscience and we also arise some questions which we wish will be interesting and simulating to them. It is our hope that their feedback and interaction will result in better understanding of these very important issues, because exploring these issues ourselves in depth is beyond our expertise.

Despite the success of (artificial) neural networks in solving many practical problems, the effort to make them a plausible model of biological neuron systems, while generating much (and often skeptical) interest, has not been very fruitful so far. One contributing fact is that the back-propagation algorithm [4] [24] [15] normally requires a dedicated feedback network to backpropagate errors, whose synapse strengthes must be kept the same as those in the original network, as suggested, for example, by Zipser and Rumelhart [27]. Although this requirement may not impose any restrictions on many applications, it does prevent the back-propagation algorithm to become a plausible model of adaptation in biological neuron systems.

Neuroscientists have always had reservations on the relevance of the adaptation algorithms to neuroscience. But on the other hand, they have discovered several possible mechanisms for biological neural systems to perform similar adaptation [10] [8] [2], for example, modulation of adaptation by means of synaptic feedback via diffusion of molecules such as Nitric Oxide and Carbon Monoxide [1] [18], reuptake of neurotransmitter and neuromodulators [5], [21], action of second messengers [12], [19], retrograde axonal transport [22] [23], and side-effects of synapse-related RNA transcription. We suggest that to whatever extent retrograde modulation of synaptic environmental exists, the ultimate impact of that modulation might be better understood by comparing the direction of that modulation with that of the locally-defined gradient exploited by our model. in the model, error propagation is no longer needed explicitly. Instead, the information needed for dendritic synapses of a neuron to adapt is the strengthes of its axonic synapses and their rates of changes. Since the axonic synapse strengthes are determined by neuron transmitters perhaps transported by microtubules [6], this information could be conveyed backwards to dendritic synapses to be used in their adaptation. Therefore, with this algorithm, all information needed for adaptation is available locally within a neuron.

At the first glance, this may seem too good to be true for networks with hidden neurons. How can a hidden neuron adapt using only local information? But in fact, as we will show, an affirmative answer can be actually found. The hint to the answer is the following fact: If a biological neuron system is working properly, then the rates of changes in the synapse strengthes in any neuron must be proportional to the corresponding errors. The information about these rates of changes is available locally in the neuron.

Our model is developed with the above idea in mind. It has the following properties related to the issue of biological plausibility.

1. The learning algorithm derived from this model does not need to backpropagate the errors. In the biological terms, this eliminates the need for a dedicated feedback network to backpropagate the errors. This makes the algorithm much more plausible in biological neuron systems.

2. All the data needed for a neuron to adapt are available locally in the neuron. These data are: the firing rates of proceeding neurons, the membrane potential of the neuron, the strengthes and their rates of changes of axonic synapses of the neuron. All those data are readily available in the neuron.

3. Using this model, the adaptation can be done concurrently at each neuron, because there is now no need to wait for errors to be propagated back. This also increases the biological plausibility of the algorithm, because it is only natural to assume that biological neurons are operated in a concurrent and parallel fashion.

4. If the same adaptation coefficient is used for all neurons, this adaptation coefficient needs only to be set at the output layer. It is cancelled in all other layers. Therefore, instead of asking every layer to set the adaptation coefficient to the same value, which is rather impossible in biological neurons, this coefficient can be set once for all at the output layer. This also makes the variation of the value of the adaptation coefficient, say, according to the magnitude of the error much easier.

5. With this model, the adaptation can be viewed as a continuous process rather than a discrete process. We will show this by using the derivatives instead of the differences in the learning algorithm. As we all know, natural processes are always continuous.

Although we think what we proposed is a plausible model for adaptation in biological neurons, we still do not know the biological mechanisms, if exist, that (1) conveys the information on axonic synapse strengthes and their rates of changes to dendritic synapses and (2) changes dendritic synapse strengthes based on the conveyed information. The investigation of such mechanisms is beyond our expertise.

3 Main results

In this section, we present our main result, a new model for adaptation of hierarchical and non-hierarchical networks. We use the following notations to specify the network configuration:

n is the label for a particular neuron;
s is the label for a particular synapse;
D_n is the set of dendritic synapses of neuron, n;
A_n is the set of (feedback-generating) axonic synapses of neuron, n;
pre_s is the presynaptic neuron corresponding to synapse, s;
$post_s$ is the postsynaptic neuron corresponding to synapse, s; and
ϵ_s indicates whether synapse, s, is excitatory (+1) or inhibitory (-1).

Therefore, for each synapse s, the triplet (pre_s, $post_s$, ϵ_s) specifies, respectively, the presynaptic neuron, the postsynaptic neuron, and whether the synapse is excitatory or inhibitory.

We use subscripts to associate variables with a particular synapse or neuron:

w_s is the strength of synapse, s;
p_n is the membrane potential of neuron, n; and
r_n is the firing rate of neuron, n.

A neuron transmits a signal from dendrites to axon. This signal flow involves generation of postsynaptic potentials following activation of a synapse by neurotransmitters, spatial and temporal integration of postsynaptic potentials at the soma, triggering of action potentials along the axon hillock, propagation of action potentials along the axon, and the release of neurotransmitters from presynaptic terminals.

The firing rate r_n of a neuron is defined as the reciprocal of the most recent interspike interval. The synapse strength w_s of a synapse is assumed to be proportional to the quantity of neurotransmitters released when a spike arrives at the synapse. It is further assumed that the short-time average of the postsynaptic potential is proportional to the product of the synapse strength w_s and the presynaptic firing rate r_{pre_s}. The membrane potential at the axon hillock p_n is thought to be an attenuated sum of the postsynaptic potentials:

$$p_n = \sum_{s \in D_n} \epsilon_s \eta_s w_s r_{pre_s} - \theta_n \tag{1}$$

where

η_s is the decremental conduction coefficient for synapse, s; and
θ_n is the offset of the membrane potential of neuron, n.

The parameter η_s represents the attenuation of the postsynaptic potential that is primarily a consequence of the distance between the synapse and the soma [14]. The firing rate r_n of the neuron is assumed to be a sigmoidal function of its membrane potential:

$$r_n = \rho_n \sigma(\lambda_n p_n) \tag{2}$$

where

$\sigma(x) = 1/(1 + e^{-x})$ is the sigmoidal function;

ρ_n is the maximal firing rate of neuron, n;

λ_n is the measure of the steepness of the sigmoidal characteristic of neuron, n.

We assume that some sort of direct feedback is available to some neurons (output neurons). Let

f_n denote the direct feedback signal for neuron, n.

If there is no direct feedback to neuron n, then $f_n = 0$. We define a feedback-generating synapse to be a synapse that does not receive the direct feedback signal. The set of such synapses is denoted by A_n, which does not include axonic synapses receiving direct feedback.

Our goal is to minimize the error given by

$$E = -\sum_n \int_0^{p_n} f_n(x)\, dx. \tag{3}$$

by adapting the synapse strengthes. Note that the above error form is a generalization of the least square error: Let $f_n(x) = t_n - x$, then

$$E = -\sum_n \int_0^{p_n} (t_n - x)\, dx = \frac{1}{2}\sum_n (t_n - p_n)^2 - \frac{1}{2}\sum_n t_n^2,$$

where

t_n is the target potential of neuron, n.

Hence the above E is equivalent to the least square error.

To minimize the error given by (3), we would like to set

$$\dot{w}_s = -\alpha_s \frac{\gamma}{\eta_s} \frac{dE}{dw_s}. \tag{4}$$

where

α_s is the postsynaptic adaptation coefficient for synapse, s;

γ is the direct feedback coefficient for all neurons.

Let us first consider adaptation in hierarchical networks.

3.1 Hierarchical networks

In a hierarchical network, a neuron may receive signals from all preceding neurons. To describe such a network, we denote the set of all neurons in the network by $N = \{n_i\}$ and introduce a partial order $\leq$ on N so that if $n_1 \leq n_2$, then there exits no axonic synapses of n_2 with n_1, that is, $D_{n_1} \cap A_{n_2} = \emptyset$. This assumption makes the "state" (i.e. the firing rate) of each neuron stable and unique.

The following learning algorithm, using only information available locally in a neuron, provides a way to minimize the error.

Theorem 1 Consider a hierarchical neural network, whose neuron have firing rates satisfy the equilibrium equations (1) and (2). The following adaptation law will ensure that the network adapt in such a way that the error E given by (3) decrease monotonically with time, that is, the equation (4) is satisfied.

$$\dot{w}_s = \alpha_s \epsilon_s r_{pre_s} (\lambda_{post_s} \phi_{post_s} \sigma(-\lambda_{post_s} p_{post_s}) + \gamma f_{post_s}), \tag{5}$$

and

$$\phi_n = \sum_{s \in A_n} \frac{\eta_s}{\alpha_s} w_s \dot{w}_s. \tag{6}$$

Proof
Omitted. ∎

The significance of this algorithm is discussed in the introduction. In particular, we have the following corollary.

Corollary 1 If $\alpha_s = \alpha$ is same for all synapses, then (5) and (6) are reduced to

$$\dot{w}_s = \epsilon_s r_{pre_s} (\lambda_{post_s} \phi_{post_s} \sigma(-\lambda_{post_s} p_{post_s}) + \alpha \gamma f_{post_s}),$$

and

$$\phi_n = \sum_{s \in A_n} \eta_s w_s \dot{w}_s.$$

Proof
Obvious. ∎

Note that if there is no direct feedback signal for a particular neuron $post_s$ ($f_{post_s} = 0$), then the adaptation coefficients α and γ are absent. In other words, the adaptation coefficients only need to be set at the output neurons.

4 Non-hierarchical networks

We now consider non-hierarchical (or recurrent) networks. We assume that the network converge to stable states, that is, the equilibrium equations (1) and (2) have at least one fixed point that is a stable attractor.

Let us label the neurons by $1, ..., K$ and the synapses by $1, ..., L$. We would like to show that the adaptation law given by the equations (5) and (6) can be used for non-hierarchical networks as well. Since the network is non-hierarchical, the equations (5) and (6) give only implicit functions of $\dot{w}_s$. To ensure the existence of these implicit functions, certain conditions must be satisfied, as stated in the following theorem.

Theorem 2 A subset of the synaptic adaptations $\{\dot{w}_s : s \in \{1, ..., M\}\}$ is implicitly defined as a function of the remaining synaptic adaptations $\{\dot{w}_s : s \in \{M+1, ..., L\}\}$ by the equations (5) and (6) if and only if the following Jacobian determinant is nonzero:

$$det(\delta_{ss'} - g_{ss'}) \neq 0, \tag{7}$$

where $(\delta_{ss'} - g_{ss'})$ is an $M \times M$ matrix with $s, s' \in \{1, ..., M\}$ and

$$\delta_{ss'} = \begin{cases} 1 & \text{if } s = s' \\ 0 & \text{otherwise} \end{cases},$$

and

$$g_{ss'} = \begin{cases} \alpha_s \epsilon_s r_{pre_s} \lambda_{post_s} \sigma(-\lambda_{post_s} p_{post_s}) \eta_{s'} w_{s'} / \alpha_{s'} & \text{if } s' \in A_{post_s} \\ 0 & \text{otherwise} \end{cases},$$

Proof

Define $h_s = \alpha_s \epsilon_s r_{pre_s} \gamma f_{post_s}$. Then by combining the equations (5) and (6), we have, for all $s \in \{1, ..., L\}$,

$$\dot{w}_s = \sum_{s'=1}^{L} g_{ss'} \dot{w}_{s'} + h_s.$$

In particular, for $s \in \{1, ..., M\}$,

$$\dot{w}_s - \sum_{s'=1}^{M} g_{ss'} \dot{w}_{s'} = h_s + \sum_{s'=M+1}^{L} g_{ss'} \dot{w}_{s'}.$$

Or

$$\sum_{s'=1}^{M} (\delta_{ss'} - g_{ss'}) \dot{w}_{s'} = h_s + \sum_{s'=M+1}^{L} g_{ss'} \dot{w}_{s'}.$$

Therefore, by the implicit function theorem, $\{\dot{w}_s\}$ is implicitly defined for $s \in \{1, ..., M\}$ if and only if

$$det(\delta_{ss'} - g_{ss'}) \neq 0.$$

■

One observation of the above theorem is the following corollary.

Corollary 2 The condition (7) is always satisfied in a hierarchical network.

Proof

For a hierarchical network, we can always label the synapses so that

$$s \leq s' \Rightarrow s' \notin A_{post_s}.$$

Therefore, $g_{ss'} = 0$ if $s \leq s'$, which implies that the matrix $(\delta_{ss'} - g_{ss'})$ is a low triangular matrix with its diagonal elements being 1. Hence,

$$det(\delta_{ss'} - g_{ss'}) = 1.$$

■

When the condition (7) is satisfied, the adaptation law given by the equations (5) and (6) will minimize the error as stated below.

Theorem 3 Consider a (non-hierarchical) neural network which has reached an equilibrium described by the equations (1) and (2). Assume that the Jacobian determinant (7) does not vanish. If synapse strengthes of the network adapt according to

$$\dot{w}_s = \alpha_s \epsilon_s r_{pre_s} (\lambda_{post_s} \sigma(-\lambda_{post_s} p_{post_s}) \sum_{s' \in A_{post_s}} \frac{\eta_{s'}}{\alpha_{s'}} w_{s'} \dot{w}_{s'} + \gamma f_{post_s}). \tag{8}$$

for $s \in \{1, ..., M\}$ and according to

$$\dot{w}_s = -\alpha_s \frac{\gamma}{\eta_s} \frac{dE}{dw_s} \tag{9}$$

for $s \in \{M+1, ..., L\}$, then the error E will decrease monotonically with time. In fact, (9) is satisfied by all synapses.

Proof

Since the condition (7) is satisfied, a unique solution to the equation (8) exists. Therefore, if we can show that $\dot{w}_s$ given by (9) satisfies the equation (8), the (9) is satisfied by all synapses, because the solution to the equation (8) is unique.

To show that, let us first take the derivative of r_i with respect to w_s in

$$r_i = \rho_i \sigma(\lambda_i p_i) = \rho_i \sigma(\lambda_i \sum_{s' \in D_i} \epsilon_{s'} \eta_{s'} w_{s'} r_{pre_{s'}} - \lambda_i \theta_i).$$

Assume $s \in D_n$, that is, $n = post_s$. Then for $i \neq n$,

$$\frac{dr_i}{dw_s} = \rho_i \sigma'(\lambda_i p_i) \lambda_i \sum_{s' \in D_i} \epsilon_{s'} \eta_{s'} w_{s'} \frac{dr_j}{dw_s} = \sum_{j=1}^{K} x_{ij} \frac{dr_j}{dw_s},$$

where $x_{ij} = \rho_i \sigma'(\lambda_i p_i) \lambda_i \sum_{s' \in A_j \cap D_i} \epsilon_{s'} \eta_{s'} w_{s'}$. For $i = n$,

$$\frac{dr_i}{dw_s} = \sum_{j=1}^{K} x_{ij} \frac{dr_j}{dw_s} + y_s,$$

where $y_s = \rho_{post_s} \sigma'(\lambda_{post_s} p_{post_s}) \lambda_{post_s} \epsilon_s \eta_s r_{pre_s}$.

Therefore, we have the following matrix equation.

$$\begin{bmatrix} x_{11}-1 & x_{12} & \dots & x_{1K} \\ x_{21} & x_{22}-1 & \dots & x_{2K} \\ \dots & & & \\ x_{K1} & x_{K2} & \dots & x_{KK}-1 \end{bmatrix} \begin{bmatrix} \frac{dr_1}{dw_s} \\ \frac{dr_2}{dw_s} \\ \dots \\ \frac{dr_K}{dw_s} \end{bmatrix} = \begin{bmatrix} 0 \\ \dots \\ -y_s \ (nth\ low) \\ \dots \\ 0 \end{bmatrix}.$$

Let

$$\begin{bmatrix} z_{11} & z_{12} & \dots & z_{1K} \\ z_{21} & z_{22} & \dots & z_{2K} \\ \dots & & & \\ z_{K1} & z_{K2} & \dots & z_{KK} \end{bmatrix} = \begin{bmatrix} x_{11}-1 & x_{12} & \dots & x_{1K} \\ x_{21} & x_{22}-1 & \dots & x_{2K} \\ \dots & & & \\ x_{K1} & x_{K2} & \dots & x_{KK}-1 \end{bmatrix}^{-1}.$$

Then

$$\begin{bmatrix} \frac{dr_1}{dw_s} \\ \frac{dr_2}{dw_s} \\ \dots \\ \frac{dr_K}{dw_s} \end{bmatrix} = \begin{bmatrix} z_{11} & z_{12} & \dots & z_{1K} \\ z_{21} & z_{22} & \dots & z_{2K} \\ \dots & & & \\ z_{K1} & z_{K2} & \dots & z_{KK} \end{bmatrix} \begin{bmatrix} 0 \\ \dots \\ -y_s \\ \dots \\ 0 \end{bmatrix} = \begin{bmatrix} -z_{1n} y_s \\ -z_{2n} y_s \\ \dots \\ -z_{Kn} y_s \end{bmatrix}.$$

In other words,

$$\frac{dr_i}{dw_s} = -z_{in} y_s.$$

Now, let us show that $\dot{w}_s$ given by (9) satisfies (8). By (3),

$$\dot{w}_s = -\alpha_s \frac{\gamma}{\eta_s} \frac{dE}{dw_s} = -\alpha_s \frac{\gamma}{\eta_s} (-\sum_{i=1}^{K} f_i \frac{dp_i}{dr_i} \frac{dr_i}{dw_s}).$$

Substituting $\frac{dp_i}{dr_i}$ and $\frac{dr_i}{dw_s}$, we have

$$\dot{w}_s = -\alpha_s \frac{\gamma}{\eta_s} \sum_{i=1}^{K} f_i \frac{1}{\rho_i \sigma'(\lambda_i p_i) \lambda_i} z_{in} y_s.$$

By the definition of y_s,

$$\dot{w}_s = -\alpha_s \epsilon_s \gamma \rho_{post_s} \sigma'(\lambda_{post_s} p_{post_s}) \lambda_{post_s} \sum_{i=1}^{K} \frac{f_i}{\rho_i \sigma'(\lambda_i p_i) \lambda_i} z_{in} r_{pre_s}.$$

Define

$$F_i = \frac{f_i}{\rho_i \sigma'(\lambda_i p_i) \lambda_i},$$

and

$$\delta_n = \delta_{post_s} = \sum_{i=1}^{K} F_i z_{in}.$$

Then

$$\dot{w}_s = -\alpha_s \epsilon_s \gamma \rho_{post_s} \sigma'(\lambda_{post_s} p_{post_s}) \lambda_{post_s} \delta_{post_s} r_{pre_s}. \tag{10}$$

By the definition of δ_{post_s},

$$\begin{bmatrix} \delta_1 \\ \delta_2 \\ \dots \\ \delta_K \end{bmatrix} = \begin{bmatrix} z_{11} & z_{12} & \dots & z_{1K} \\ z_{21} & z_{22} & \dots & z_{2K} \\ \dots & & & \\ z_{K1} & z_{K2} & \dots & z_{KK} \end{bmatrix}^T \begin{bmatrix} F_1 \\ F_2 \\ \dots \\ F_K \end{bmatrix}.$$

By the definition of z_{ij},

$$\begin{bmatrix} \delta_1 \\ \delta_2 \\ \dots \\ \delta_K \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1K} \\ x_{21} & x_{22} & \dots & x_{2K} \\ \dots & & & \\ x_{K1} & x_{K2} & \dots & x_{KK} \end{bmatrix}^T \begin{bmatrix} \delta_1 \\ \delta_2 \\ \dots \\ \delta_K \end{bmatrix} - \begin{bmatrix} F_1 \\ F_2 \\ \dots \\ F_K \end{bmatrix}.$$

In other words,

$$\delta_{post_s} = \sum_{i=1}^{K} x_{in} \delta_i - F_{post_s} = \sum_{i=1}^{K} \rho_i \sigma'(\lambda_i p_i) \lambda_i \sum_{s' \in A_{post_s} \cap D_i} \epsilon_{s'} \eta_{s'} w_{s'} \delta_i - F_{post_s}.$$

Substituting F_{post_s} and δ_{post_s} back to (10), we have

$$\begin{aligned} \dot{w}_s &= -\alpha_s \epsilon_s \gamma \rho_{post_s} \sigma'(\lambda_{post_s} p_{post_s}) \lambda_{post_s} r_{pre_s} \\ &\quad \sum_{i=1}^{K} \rho_i \sigma'(\lambda_i p_i) \lambda_i \sum_{s' \in A_{post_s} \cap D_i} \epsilon_{s'} \eta_{s'} w_{s'} \delta_i + \alpha_s \epsilon_s \gamma r_{pre_s} f_{post_s} \\ &= -\alpha_s \epsilon_s \gamma \rho_{post_s} \sigma'(\lambda_{post_s} p_{post_s}) \lambda_{post_s} r_{pre_s} \\ &\quad \sum_{s' \in A_{post_s}} \rho_{post_{s'}} \sigma'(\lambda_{post_{s'}} p_{post_{s'}}) \lambda_{post_{s'}} \epsilon_{s'} \eta_{s'} w_{s'} \delta_i + \alpha_s \epsilon_s \gamma r_{pre_s} f_{post_s}. \end{aligned}$$

On the other hand,

$$\dot{w}_{s'} = -\alpha_{s'} \epsilon_{s'} \gamma \rho_{post_{s'}} \sigma'(\lambda_{post_{s'}} p_{post_{s'}}) \lambda_{post_{s'}} \delta_{post_{s'}} r_{pre_{s'}};$$

$$\rho_{post_s} \sigma'(\lambda_{post_s} p_{post_s}) \lambda_{post_s} = \sigma(-\lambda_{post_s} p_{post_s}) r_{post_s};$$

and

$$r_{post_s} = r_{pre_{s'}}.$$

Therefore,

$$\dot{w}_s = \alpha_s \epsilon_s r_{pre_s} (\lambda_{post_s} \sigma(-\lambda_{post_s} p_{post_s}) \sum_{s' \in A_{post_s}} \frac{\eta_{s'}}{\alpha_{s'}} w_{s'} \dot{w}_{s'} + \gamma f_{post_s}).$$

■

5 Conclusion

We proposed a learning algorithm based on a new model that requires no explicit error back-propagation. Using the algorithm, information needed for adaptation is available within a neuron. We proved the correctness of the algorithm for both hierarchical and non-hierarchical networks. For non-hierarchical networks, certain Jacobian condition must be satisfied to ensure the existence of the solution to the implicit adaptation law. We believe that this adaptation law has the potential to model the biological adaptation taking place in biological neural systems.

References

[1] M. Barinaga (1991). Is nitric oxide the retrograde messenger. *Science 254*, pp. 1296-1297.

[2] T. H. Brown, A. H. Ganong, E. W. Kairiss, C. L. Keenan and S. R. Kelso (1989). Longterm potentiation in two synaptic systems of the hippocampal brain slice. In *Neural Models of Plasticity* (J. H. Byrne and W. O. Berry, eds.), Academic Press, 1989, pp. 266-306.

[3] T. H. Brown, E. W. Kairiss, and C. L. Keenan (1990). Hebbian synapses: biophysical mechanisms and algorithms. *Annual Review of Neuroscience, 13*, pp. 475-511.

[4] A. E. Bryson and Y.-C. Ho (1969). *Applied Optimal Control.* New York: Blaisdell.

[5] P. De Camilli and R. Jahn (1991). Pathways to regulated exocytosis in neurons. *Annual Review of Physiology, 52*, pp. 624-645.

[6] J. E. Dayhoff, S. R. Harmeroff, C. E. Swenberg, R. Lahoz-Beltra, and A. Samsonovich (1992). Biological Learning with Cytoskeletal Signaling. *Proceedings of the 1992 IEEE/INNS International Joint Conference on Neural Networks, Vol. 2*, pp. 45-50.

[7] D. Gardner (1993). *The Neurobiology of Neural Networks*, The MIT Press.

[8] R. D. Hawkins, E. R. Kandel and S. A. Siegelbaum (1993). Learning to modulate transmitter release: themes and variations in synaptic plasticity. *Annual Review of Neuroscience, 16*, pp. 625-665.

[9] J. Hertz, A. Krogh and R. G. Palmer (1991). *Introduction to the Theory of Neural Computation*, Addison-Wesley.

[10] D. V. Madison, R. C. Malenka and Nicoll (1991). Mechanisms underlying longterm potentiation of synaptic transmission. *Annual Review of Neuroscience, 14*, pp. 379-397.

[11] B. W. Mel (1992). NMDA-based pattern discrimination in a modeled cortical neuron. *Neural Computation, 4(4)*, pp. 502-517.

[12] D. Piomelli, D. Volterra, N. Dale, S. A. Siegelbaum, E. R. Kandel, J. H. Schwartz and F. Belardetti (1987). Lipoxygense metabolites of arachidonic acid as second messengers for presynaptic inhibition of Aplysia sensory cells. *Nature, 328*, pp. 38-43.

[13] K. H. Pribram (1993) *Rethinking Neural Networks: Quantum Fields and Biological Data*, Lawrence Erlbaum Associates, Publishers.

[14] W. Rall (1977). Core conductor theory and cable properties of neurons. In *Handbook of Physiology, Vol. 1* (J. M. Brookhart and V. B. Mountcastle, eds.), Williams and Wilkins, Baltimore, 1977, pp. 39-97.

[15] D. E. Rumelhart, G. E. Hinton, and G. E. Williams (1986). Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol 1: Foundations* (D. E. Rumelhart and J. L. McClelland, eds.), MIT Press, Cambridge, 1986, pp. 318-362.

[16] N. A. Schmajuk and J. D. DiCarlo (1992). Stimulus configuration, spatial learning, and hippocampal function. *Psychological Review, 99(2)*, pp. 268-305.

[17] N. A. Schmajuk and H. T. Blair (1993). Stimulus configuration, classical conditioning, and hippocampal function. *Behavioral Brain Research*, submitted.

[18] E. M. Schuman and D. V. Madison (1991). A requirement for the intercellular messenger nitric oxide in long term potentiation. *Science 254*, pp. 1503-1506.

[19] J. H. Schwartz and T. R. Scoderling (1983). Molecular mechanisms for memory: second messenger induced modifications of protein kinase in nerve cells. *Annual Review of Neuroscience, 7*, pp. 291-301.

[20] G. Tesauron (1990). Neural models of classical conditioning: a theoretical viewpoint. In *Connectionist Modeling and Brain Function: The Developing Interface* (S. J. Hanson and C. R. Olson, eds.), MIT Press, Cambridge, 1990, pp. 74-104.

[21] F. Torri-Tarelli, A. Villa, F. Valtorta, P. De Camilli, P. Greengard and B. Ceccarelli (1990). Redistribution of synaptophysin and synapsin I during α-latrotoxin-induced release of neuro-transmitter at the neuronmusular junction. *Journal of Cell Biology, 110*, pp. 449-459.

[22] W. S. Trimble, M. Linial, and R. H. Richter (1991). Cellular and molecular biology of the presynaptic nerve terminal. *Annual Review of Neuroscience, 14*, pp. 93-122.

[23] R. B. Vallee and G. S. Bloom (1991). Mechanisms of fast and slow axonal transport. *Annual Review of Neuroscience, 14*, pp. 59-92.

[24] P. J. Werbos (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences.* Ph.D. Thesis, Harvard University.

[25] J. H. Williams, M. L. Errington, M. A. Lynch, and T. V. P. Bliss (1989). Arachidonic acid induces a long-term activity-dependent enhancement of synaptic transmission in the hippocampus. *Nature 341*, pp. 739-642.

[26] R. E. Zigmond, M. A. Schwarzschild, and A. R. Rittenhouse (1989). Acute regulation of tyrosine hydroxylase by nerve activity and by neurotransmitters via phosphorylation. *Annual Review of Neuroscience, 12*, pp. 415-461.

[27] D. Zipser and D. E. Rumelhart (1990). Neurobiological significance of new learning models. In *Computational Neuroscience* (E. Schwartz, eds.), MIT Press, Cambridge, 1990, pp. 192-200.

A Dynamic Model for Winner-Take-All Competition

Coskun Baysoy & Larry L. Kinney

Department of Electrical Engineering
University of Minnesota
Minneapolis, MN 55455

Abstract

This paper presents a simple, continuous dynamic model for so-called winner-take-all competition. The model allows persistent external inputs and is fast. No resetting of the model state is required and the dynamic range of the model state, depending on the range of the external inputs, can be arbitrarily large. The model has the property that it not only decodes the highest external input, but also passes its real value onto the outputs. This is a very desirable property especially in combinatorial optimization applications.

I. Introduction

A competitive dynamical system is called a *winner-take-all (WTA)* system if its dynamics are convergent and any of its stable equilibrium states has only one non-zero component [1]-[2], [14]. Depending on whether the system has been furnished with external inputs, the dynamics are typically designed such that the non-zero component of the equilibrium state indicates either the largest component of the initial state vector or the external input vector. In either case, the system functions as a maximum decoder — an important subsystem centrally used by all neural competitive learning schemes [1-6]. WTA systems have also applications in pattern recognition [1], [3], [8] and combinatorial optimization [9].

A large number of WTA models exists in the literature. Static models consist of the selection of the largest valued input signal followed by thresholding [5]. For example, a single, static max neuron of the type proposed in [7] may be used to implement the maximum selector. In some earlier dynamic WTA models [1], [3], [10-13] no inputs were considered. Such models decode the largest of the initial states and were originally designed to choose a winner from noisy data. However, such WTA systems need to be reset for each new input data, which makes them less attractive if they have to operate within a larger continuous dynamical system.

Several dynamic WTA models with external inputs have later been proposed [1], [5-6], [15]. Recently, Kaski & Kohonen [6] further classified the latter models into two categories: *Strong* models versus *weak* models. In strong WTA models, external inputs override the background noise activity (inherent in real systems) and decode the largest external input independent of the initial state of the system. A weak WTA model, which lacks this property, must always be reset to a uniform initial state before the application of a new input. The WTA models proposed in [5-6] and [15] are weak models. The correct operation of a weak WTA model is crucially dependent on the uniformity of the initial state vector; and just for this reason, it is unrealistic to assume that such systems would work in a noisy real operating environment. An automatic resetting mechanism for weak WTA systems was proposed in [5] and claimed to guarantee an uniform initial state [6]. Still, the operation of a weak system with automatic reset is not truly continuous in that it operates in cycles (a new input is accepted and processed in each cycle). In other words, such a system operates only on discrete-time input signals. Finally, there exist

numerous electronic WTA circuits published in the literature [18-23], [13]. However, often these papers fall short of presenting a sound dynamic model in terms of the circuit parameters and variables.

In this paper, we propose a convergent WTA network modeled as a continuous dynamical system with constant but real external inputs. This network was originally designed by us as a maximum-selector which resulted in efficient VLSI implementations of max/min neurons [16]. It is now claimed that it is also a maximum decoder, i.e., a WTA system, and that it is a strong model with an arbitrarily large dynamic range. Moreover, the proposed WTA system has the desirable property that it not only decodes the highest external input, but also passes its value on to the output. In other words, every equilibrium state of the system is asymptotically stable and has only one non-zero unit which indicates the highest external input and also possesses its numerical value.

II. Description and Analysis of the WTA Model

The network has n neurons and n inputs. Associated with the kth neuron, there exists a real variable $x_k(t)$ representing the state of the kth neuron at time t and an external input u_k. The feature states of the network is determined by the system of first-order nonlinear equations

$$\frac{dx_k}{dt} = -x_k + S(u_k - \sum_{i \neq k}^{n} x_i) \qquad k = 1,2,...,n \tag{1}$$

where $S(.)$ is the threshold logic function, i.e.,

$$S(a) = Max\{a,0\} \tag{2}$$

Let $\boldsymbol{u} = (u_1,u_2,...,u_n)$ and $\boldsymbol{x} = (x_1,x_2,...,x_n)$ denote the external input and state vectors, and let

$$z(t) = \sum_{i=1}^{n} x_i(t) \tag{3}$$

and

$$u_{max} = Max\{u_1,u_2,...,u_n\} \tag{4}$$

The discussion in the rest of this section relates to the analysis of (1) with respect to non-negative persistent external inputs $\boldsymbol{u}$, thus $u_i \geq 0$ is time-invariant $\forall k$, and it is also assumed that the difference between u_{max} and the second highest input is not zero. The case in which two or more inputs are equal will be analyzed later. In what follows, we show that, starting from an *arbitrary* initial state $\boldsymbol{x}(0)$, as $t \to \infty$, the state vector converges to the only existing equilibrium state, i.e., $\boldsymbol{x}(t) \to \boldsymbol{x}^*$ and $z(t) \to z^*$, in which the following is satisfied by all x_k^*

$$x_k^* = \begin{cases} u_{max} & if \quad u_k = u_{max} \\ 0 & if \quad u_k < u_{max} \end{cases} \tag{5}$$

With the assumptions above, Eq. (5) implies that $z^* = u_{max}$ is also satisfied. In other words, the network processes the information on its external inputs and decodes the highest valued input.

Remark 1 Eqs. (1) are of the form $dx_k / dt = g_k(\boldsymbol{x})$. It can be verified that the functions g_k satisfy $\partial g_k / \partial x_j \le 0$, $\forall j$. Thus, (1) is a so-called *competitive* dynamical system [1-2], i.e., an increase in the value of x_j cannot increase the rate of change dx_k / dt in any other neuron state x_k (including x_j).

2.1 Dynamic Range of the WTA Network

In this subsection we are concerned with the transient and ultimate (steady-state) behavior of the solutions of the WTA model. In particular, we ask: Are all solutions of the system of equations (1) bounded as $t \to \infty$? It is necessary to investigate such bounds to ensure proper operation of any hardware implementation of (1).

Lemma 1 The states $x_k(t)$ are bounded for all $t \ge 0$ by constants M_k, i.e., $|x_k(t)| \le M_k$, and the bounds can be computed explicitly as follows

$$M_k = |x_k(0)| + S(u_k - \sum_{i \ne k} c_i(0)) \tag{6}$$

where $c_i(0) = 0$ if $x_i(0) \ge 0$; $c_i(0) = x_i(0)$, otherwise.

Proof The general solution of the *kth* equation in (1) is of the form

$$x_k(t) = x_k(0)e^{-t} + \int_0^t e^{-(t-s)} S(u_k - \sum_{i \ne k} x_i(s))\, ds \tag{7}$$

The second term in (7) is a non-negative quantity for all $t \ge 0$ because $S(.) \ge 0$. Thus, we have the lower bounds

$$x_k(t) \ge \min_t \{x_k(0)e^{-t}\}, \quad \forall k \tag{8}$$

Furthermore, it follows from (7) that

$$|x_k(t)| \le |x_k(0)e^{-t}| + \int_0^t e^{-(t-s)} S(u_k - \sum_{i \ne k} x_i(s))ds \tag{9}$$

$$\le |x_k(0)| + \underset{t}{Max}\left\{ S(u_k - \sum_{i \ne k} x_i(t)) \right\} \int_0^t e^{-(t-s)} ds$$

$$\le |x_k(0)| + (1 - e^{-t}) S(\underset{t}{Max}\{u_k - \sum_{i \ne k} x_i(t)\}) \tag{10}$$

$$\le |x_k(0)| + S(u_k - \underset{t}{Min}\{\sum_{i \ne k} x_i(t)\}) \tag{11}$$

Since $x_k(t)$, $\forall k$, are bounded from below for all $t \ge 0$ by (8), there exists a finite number M_k independent of t such that $|x_k(t)| \le M_k$ for all $t \ge 0$. The proof is completed by considering inequalities (8) and (11) simultaneously.

Remark 3 Notice in (7) that the transient solution (the first term) approaches zero as $t \to \infty$. That is, in the limit as $t \to \infty$, the behavior of the solution is solely governed by that of its steady-state part (the second term), and from (9), the following inequality is satisfied

$$|x_k(t \to \infty)| \le S(u_k - \underset{t}{Min}\{\sum_{i \ne k} x_i(t \to \infty)\}) \tag{12}$$

Also, from (8) we have the lower bounds $x_k(t \to \infty) \ge 0, \ \forall k$. Using this information, it then follows from (12) that $0 \le x_k(t \to \infty) \le S(u_k)$, or equivalently

$$0 \le x_k(t \to \infty) \le u_k, \ \forall k \tag{13}$$

since $u_k \ge 0, \ \forall k$. Thus, after the transients have decayed to zero, the neuron states are limited to a real interval defined by their present inputs. Particularly, (13) says that if $u_k = 0$, then, as $t \to \infty$, $x_k(t) \to 0$ for any initial state $x_k(0)$. This implies that if the external input vector is the all-zero vector, then the WTA network cannot make a selection since there is nothing to select.

2.2 Locations of Equilibrium States

The current results imply that the WTA network is essentially *stable* in the sense that solutions may oscillate but never grow arbitrarily as $t \to \infty$. For our purposes, however, the network must not oscillate but converge to an equilibrium state and stay there indefinitely, or as long as the inputs remain unchanged. We defer the detailed stability analysis of the network to the next two subsections. In the following, we prove that, under the single maximum assumption, i.e., $u_k = u_{max} > u_i, \ \forall i \ne k$, the WTA network has a *unique* equilibrium state which satisfies (5).

Constant solutions $\boldsymbol{x}^*$ of Eqs. (1) are the equilibrium states of the WTA network which satisfy $dx_k^*/dt = 0, \ \forall k$, or equivalently

$$-x_k^* + S(u_k - z^* + x_k^*) = 0, \ \forall k \tag{14}$$

where z^* is the sum of all neuron states x_i^* as defined with (3). Now, we are ready to state the main result of this subsection.

Theorem 1 If $u_k = u_{max} > u_i, \ \forall i \ne k$, then the WTA network has a unique equilibrium state $\boldsymbol{x}^*$ and $\boldsymbol{x}^*$ is given by $x_k^* = u_{max}$ and $x_i^* = 0$ for all $i \ne k$.

Proof With the substitution of (2) into (14), the *ith* equation in (14) can be written as

$$x_i^* = Max\{u_i - z^* + x_i^*, \ 0\} \tag{15}$$

Equality (15) is satisfied if and only if at least one of the following two conditions is satisfied:

i) $x_i^* \ge 0$ and $z^* = u_i$ (16)

ii) $x_i^* = 0$ and $z^* \ge u_i$ (17)

From (16) and (17), an immediate result is that $z^* \geq u_i$, $\forall i$. This, of course, implies $z^* \geq u_{max}$. However, $z^* > u_{max}$ is not possible if $u_{max} > 0$. This simply follows from the contradiction that $z^* > u_{max} > 0$ requires $x_i^* = 0$, $\forall i$, in order to satisfy (15) for all i, but if $x_i^* = 0$, $\forall i$, then we have $z^* = 0$. Therefore, $z^* = u_{max}$ must be satisfied at an equilibrium state of (1). Since $z^* > u_i$ for all $u_i < u_{max}$, it then follows from (17) that $x_i^* = 0$ for all $i \neq k$. On the other hand, clearly $z^* = x_k^* = u_k = u_{max}$. This completes the proof.

Remark 4 If the WTA network is convergent, the above theorem says that the network will not only decode but also select the highest input provided that the highest input is unique.

If there are two or more external inputs sharing the same value, then the equilibrium state is not unique. Suppose that the inputs u_k and u_j satisfy $u_k = u_j = u_{max}$. Then, it can be shown that $x_i^* = 0$ if $u_i < u_{max}$ (or equivalently, if $i \neq k, j$). However, x_k^* and x_j^* cannot be determined by the dynamics of the WTA network alone because their values also depend on the initial states $x_i(0)$. Note that an exception case is treated in *Lemma 2* below. On the other hand, it is easy to see that x_k^* and x_j^* take on values on the hypersurface defined by

$$x_k^* + x_j^* = u_{max} \tag{18}$$

This result can be extended to the case in which more than two inputs share the same value.

Remark 5 The above analysis points out that when the equilibrium state is not unique (i.e., $u_k = u_{max} > u_i$, $\forall i \neq k$ is not satisfied), the WTA network has an infinite number of equilibrium states which are not simple (all connected) and lie on a hypersurface in the state-space.

The exact locations of the equilibrium states when there are two or more equal inputs will be estimated in the next subsection. However, we examine a special case in the following lemma.

Lemma 2 Assume that m of the n external inputs have *exactly* the same (maximum) value u_{max}. Then, if the initial state vector is uniform ($x_i(0) = c$, $\forall i$, where c is a real number), then

$$x_i^* = \frac{u_{max}}{m} \quad \text{if } u_i = u_{max}$$
$$x_i^* = 0 \quad \text{if } u_i < u_{max}$$

Proof The following proof is conditioned on the convergence of the network (The proof that the WTA system is convergent will be given in a later subsection). Suppose that $u_k = u_j = u_{max}$. Consider the states $x_k(t)$ and $x_j(t)$, and their general solutions (from (7))

$$x_k(t) = ce^{-t} + \int_0^t e^{-(t-s)} S(u_{max} - z(s) + x_k(s))\, ds \tag{19}$$

$$x_j(t) = ce^{-t} + \int_0^t e^{-(t-s)} S(u_{max} - z(s) + x_j(s))\, ds \tag{20}$$

where $z(t)$ was previously defined (Eq. (3)). If the network is convergent, and since Eqs. (19) and (20) are basically the same, as $t \to \infty$, we have $e(t) = [x_k(t) - x_j(t)] \to 0$. In other words,

in the limit $x_{com} = x_k^* = x_k(t \to \infty) = x_j^* = x_j(t \to \infty)$ is satisfied. Since this inequality holds for all x_k^*, x_j^* such that $u_k = u_j = u_{max}$, and from Thm. 1, $x_i^* = 0$ if $u_i < u_{max}$, we now have

$$z^* = u_{max} = \sum_{i=1}^{n} x_i^* = m\, x_{com} \tag{21}$$

Therefore, $x_{com} = u_{max} / m$.

Remark 6 Notice that $m = 1$ case was already treated in Thm. 1 and recovered in the above lemma. This lemma distinguishes yet another situation where the equilibrium state is unique, i.e., in the sense that it is determined by the dynamics of the network alone. Note that, given the structure of the network, the dynamics can only be modified by the external input vectors. That is, different initial states do not change the dynamics.

It may be desirable to select only one of the inputs as the winner if there are two or more inputs sharing the exact same value. A modified WTA network which achieves this will be described briefly later in Sec. 3.

Remark 7 It is important to notice that the WTA network produces a unique winner even if the difference between the highest and the next highest input is arbitrarily small (but not zero). In [5], Kohonen mentions that in real competitive learning processes, the influence of having several winners is often negligible; and in fact, sometimes useful. On the other hand, in a real environment, it is quite unlikely that any two of the inputs will have exactly the same value.

2.3 Local Stability of Equilibrium States

If the WTA network is initially in an equilibrium state $\boldsymbol{x}^*$, it remains there forever. Here, we ask and answer the question: If the network is initially at a state $\boldsymbol{x(0)}$ sufficiently near $\boldsymbol{x}^*$, does the associated solution $\boldsymbol{x}(t)$ have the property that $\boldsymbol{x}(t) \to \boldsymbol{x}^*$ as $t \to \infty$? The stability study of any given equilibrium state $\boldsymbol{x}^*$ can be reduced to the stability study of the origin as follows: Set $x_k(t) = y_k(t) + x_k^*$ in (1) and (3) to obtain

$$\frac{dy_k}{dt} = \tilde{g}_k(\boldsymbol{y}) = -y_k - x_k^* + S\Big(u_k - u_{max} + x_k^* - \sum_{i \neq k}^{n} y_i\Big), \quad \forall k \tag{22}$$

Since $\tilde{g}_k(\boldsymbol{0}) = 0$ for all k, $\boldsymbol{y} = \boldsymbol{0}$ is an equilibrium state of (22) and it will be stable or unstable in the same manner that $\boldsymbol{x} = \boldsymbol{x}^*$ is for (1). Consider now the approximation $d\boldsymbol{y}/dt = \boldsymbol{G}\boldsymbol{y} + \boldsymbol{h}(\boldsymbol{y})$ to (22) around $\boldsymbol{y} = \boldsymbol{0}$, where $\boldsymbol{G} = [G_{kj}] = [\partial \tilde{g}_k(\boldsymbol{0}) / \partial y_j]$ is the Jacobian evaluated at $\boldsymbol{0}$ where

$$G_{kj} = \begin{cases} -1 & \text{if } j = k \\ 0 & \text{if } j \neq k \text{ and } u_k < u_{max} \\ -1 & \text{if } j \neq k \text{ and } u_k = u_{max} \end{cases} \tag{23}$$

and $\boldsymbol{h}(\boldsymbol{y})$ contains the higher order terms and satisfies $\boldsymbol{h}(\boldsymbol{0}) = \boldsymbol{0}$ and $|\boldsymbol{h}(\boldsymbol{y})|/|\boldsymbol{y}| \to 0$ as $|\boldsymbol{y}| \to 0$. Since the contribution of $\boldsymbol{h}(\boldsymbol{y})$ around $\boldsymbol{y} = \boldsymbol{0}$ is insignificant, we may study the linear system

$dy/dt = Gy$ alone to determine the stability of (22) around $y = 0$. The system $dy/dt = Gy$, with the substitution of (23), can be written explicitly as, $\forall k$,

$$\frac{dy_k}{dt} = \begin{cases} -y_k & \text{if } u_k < u_{max} \\ -\sum_{j=1}^{n} y_j & \text{if } u_k = u_{max} \end{cases} \tag{24}$$

Consider now the case in which m of the n external inputs u_i are equal to u_{max}. Hence, $n-m$ equations in (24), which correspond to the cases in which $u_i < u_{max}$, are of the decoupled form $dy_i/dt = -y_i$ and have the property that $y_i(t) \to 0$ as $t \to \infty$. Substituting these $n-m$ known solutions into the remaining m equations, (24) can be rewritten as $d\hat{y}(t)/dt = A\hat{y}(t) + f(t)$, where y_k is a component of the variable vector $\hat{y}$ if and only if $u_k = u_{max}$, and A is the $m \times m$ matrix of -1's. Furthermore, since $f(t) \to 0$ as $t \to \infty$, it suffices to study $d\hat{y}/dt = A\hat{y}$ alone to draw conclusions about the stability of the origin, whose solutions are of the form:

$$\hat{y}(t) = \begin{bmatrix} e^{-mt} & -1 & -1 & & -1 \\ e^{-mt} & 1 & 0 & \cdots & 0 \\ e^{-mt} & 0 & 1 & & 0 \\ & \vdots & & \ddots & \vdots \\ e^{-mt} & 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_m \end{bmatrix} \tag{25}$$

where constants c_k, which are dependent on the initial states, are given by

$$c_i = \begin{cases} \frac{1}{m}\sum_{k}^{m} \hat{y}_k(0) & \text{for } i = 1 \\ \hat{y}_k(0) - \frac{1}{m}\sum_{k}^{m} \hat{y}_k(0) & \text{for } i > 1 \end{cases} \tag{26}$$

Recall that y_k appears in $\hat{y}$ iff $u_k = u_{max}$, and that $y_i(t) \to 0$ if $u_i < u_{max}$. By substituting proper y_k's into (25), (26), we conclude that, if system (22) starts with $y_k(0) = \sigma_k$, $\forall k$, where σ_k is a real number sufficiently close to 0, then in the limit as $t \to \infty$, we have $y_k(t) \to y_k^*$ such that

$$y_k^* = y_k(0) - (1/m) \sum_{(u_i = u_{max})} y_i(0) \qquad \text{if } u_k = u_{max} \tag{27}$$

$$y_k^* = 0 \qquad \text{if } u_k < u_{max}$$

In other words, system (22) may not return to the zero state.

Remark 8 From (27), local solutions of (22) do not exhibit any bizarre behavior as $t \to \infty$, but approach some finite constant points in the state-space. This, as mentioned before, implies that local solutions of the original system (1), or the WTA network, approach constant vectors as $t \to \infty$ and stay there thereafter (or at least until the new inputs are applied).

Nevertheless, contrary to our initial expectation, in response to a small initial perturbation from $\boldsymbol{y} = \boldsymbol{0}$, system (22) does not necessarily return to $\boldsymbol{0}$ (while it certainly can), but it might converge to some point $\boldsymbol{y}^*$ given by (27). Such points, however, always satisfy $\sum_{i=1}^{n} y_k^* = 0$, which implies that

$$\sum_{i=1}^{n} x_k^* = u_{max} \tag{28}$$

in terms of the equilibrium states of the original system (1). This result can be proven with the help of Thm. 1. Furthermore, with the proper variable substitutions, a formula similar to (27) can be obtained for the equilibrium states of the original system as follows:

$$x_k^* = x_k(0) - \frac{1}{m} \sum_{\substack{i \\ (u_i = u_{max})}}^{m} x_i(0) + \frac{u_{max}}{m} \qquad \text{if } u_k = u_{max} \tag{29}$$

$$x_k^* = 0 \qquad \text{if } u_k < u_{max} \tag{30}$$

Remark 9 Notice that the summation in (29) is over the initial states $x_i(0)$ of those neurons whose inputs satisfy $u_i = u_{max}$. It can be seen that Eqs. (29) and (30) verify the result stated in Lemma 2. Moreover, they point out that in response to a small but *uniform* perturbation from the equilibrium state $\boldsymbol{x}^*$, the WTA network always converges back to $\boldsymbol{x}^*$.

We remind here that the above analysis is intended to be valid only within a sufficiently small neighborhood of an equilibrium state. Furthermore, we specifically treated the case in which the network was assumed to have two or more inputs that are equal to u_{max}. More specifically, it was assumed that m external inputs have precisely the same value u_{max}. If $m = 1$, there is a unique equilibrium state (Thm. 1) which is asymptotically stable.

2.4 A Global Property of the WTA Network

The preceding results are local in nature which, in general, cannot predict the global behavior of $\boldsymbol{x}(t)$ for an arbitrary $\boldsymbol{x}(0)$. In this subsection, however, we claim that local properties also carry globally. That is, for any $\boldsymbol{x}(0)$, there exists a solution $\boldsymbol{x}(t)$ which has the property that $\boldsymbol{x}(t) \to \boldsymbol{x}^*$ as $t \to \infty$, where $\boldsymbol{x}^*$ is an equilibrium state of the WTA network.

Reconsider the system of differential equations (1) that describes the WTA network. Define the new variables η_k by $\eta_k = u_k - \sum_{i \neq k}^{n} x_i$. Then, system (1) can be written in terms of η_k's as

$$\frac{d\eta_k}{dt} = -\eta_k - \sum_{i \neq k}^{n} S(\eta_i) + u_k, \quad \forall k \tag{31}$$

Notice that this coordinate transformation allowed us to write the threshold logic function of the sum appearing in (1) as the sum of the threshold logic functions of the individual neuron states. System (31) is a special case of the well-known *additive* neural network model. System (1), on the other hand, may be considered as a variant of the so-called *brain-state-in-a-box* model (see, e.g., [1] for their review).

There is a known Liapunov function $V(.)$ for the additive model in the literature discovered by Cohen and Grossberg [17], which when applied to system (32) yields

$$\frac{dV}{dt} = -\sum_{k=1}^{n} S'(\eta_k)\left[-\eta_k - \sum_{i \neq k}^{n} S(\eta_i) + u_k\right]^2 \qquad (32)$$

Since $S'(\eta_k) \geq 0$ $(0 \leq S'(0) \leq 1)$ always holds, $dV / dt \leq 0$ is satisfied for all t.

III. Concluding Remarks

A simple and fast continuous dynamic winner-take-all competition model with external inputs has been presented in this paper. It has been shown that every equilibrium state of the system is stable and has only one non-zero component which not only indicates the largest component of the (persistent) external input pattern (a maximum decoder function), but also possesses its numerical value (a maximum selector function).

We are not aware of any other winner-take-all competition model which is also a maximum selector. Maximum selectors are useful in, e.g., optimization algorithms. The proposed model puts no restriction on the magnitudes of the external input signals and no resetting of the states is needed. Also, dynamic range of the model state is always at least as wide as that of the external input signals.

The model always flags only one "winner" if the difference between the highest and the second highest input signal is not zero (it may be arbitrarily small). Although it is quite unlikely (in a real environment), if there are m equal inputs at the highest value, there will be m winners. We note here that the following modification to the original model ensures only one winner under all circumstances:

$$\frac{dx_k}{dt} = -x_k + S\left(u_k - \lambda \sum_{i \neq k}^{n} x_i\right), \qquad k = 1,2,\ldots,n$$

The only difference between this and the original model (1) is the parameter λ multiplying the summation above. The value of λ is given by $\lambda = 1 + \varepsilon$, where $\varepsilon > 0$ is a small real number, possibly with an upper bound. Current work is further analyzing these systems and exploring ways to utilize them in 0-1 integer programming.

References

[1] Grossberg, S., "Nonlinear Neural Networks: Principles, Mechanisms, and Architectures," *Neural Networks*, Vol. 1, pp. 17-61, 1988.

[2] Hirsch, M. W., "Convergent Activation Dynamics in Continuous Time Networks," *Neural Networks*, Vol. 2, pp. 331-349, 1989.

[3] Lippmann, R. P., "An Introduction to Computing with Neural Nets," *IEEE ASSP Magazine*, pp. 4-22, Apr. 1987.

[4] Kohonen, T., "An Introduction to Neural Computing," *Neural Networks*, Vol. 1, pp. 3-16, 1988.

[5] Kohonen, T., "Physiological Interpretation of the Self-Organizing Map Algorithm," *Neural Networks*, Vol. 6, pp. 895-905, 1993.
[6] Kaski, S. and T. Kohonen, "Winner-Take-All Networks for Physiological Models of Competitive Learning," Neural Networks, Vol. 7, Nos. 6/7, pp. 973-984, 1994.
[7] Baysoy, C. and L. L. Kinney, "Subthreshold MOS Fuzzy Max/Min Neuron Circuits," *Proc. World Congress on Neural Networks*, San Diego, Vol. II, pp. 500-505, 1994.
[8] Carpenter, G. A., "Neural Network Models for Pattern Recognition and Associative Memory," *Neural Networks*, Vol. 2, pp. 243-257, 1989.
[9] Takefuji, Y., K-C Lee, and H Asio, "An Artificial Maximum Neural Network: A Winner-Take-All neuron Model Forcing the State of the System in A Solution Domain," *Biol. Cybern.*, **67**, pp. 243-251, 1992.
[10] Grossberg, S., "Contour Enhancement, Short Term Memory, and Constancies in Reverberating Neural Networks," *Studies in Applied Mathematics*, L11, pp. 213-257.
[11] Majani, E., R. Erlanson, and Y. Abu-Mostafa, "On the K-Winners-Take-All Network," in *Advances in Neural Information Processing Systems I*, Touretzky D. S. (Ed.), Morgan Kaufmann, San Mateo, CA, pp. 634-642, 1989.
[12] Floreen, P., "The Convergence of Hamming Memory Networks," *IEEE Trans. Neural Networks*, Vol. 2, No. 4, pp. 449-457, July 1991.
[13] Seiler G. and J. A. Nossek, "Winner-Take-All Cellular Neural Networks," *IEEE Trans. Circuits & Systems -II: Analog and Digital Signal Processing*, Vol. 40, No, 3, pp. 184-190, March 1993.
[14] Feldman, J. A. and D. H. Ballard, "Connectionist Models and Their Properties," *Cognitive Science*, **6**, pp. 205-254, 1982.
[15] Wolfe, W. J. et al., "K-Winner Networks," *IEEE Trans. Neural Networks*, Vol. 2, No. 2, pp. 310-315, March 1991.
[16] Baysoy C. and L. L. Kinney, "Asymmetric Feedback Max/Min Circuits of O(n) Complexities for Analog VLSI Neural, Multiple-Valued, and Fuzzy Logic Systems," *Proc. World Congress on Neural Networks*, Portland, Vol. II, pp. 97-101, July 1993.
[17] Cohen, M. A. and S. Grossberg, "Absolute Stability of Global Pattern Formation and Parallel Memory Storage by Competitive Neural Networks," IEEE Trans. SMC-13, pp. 815-826, 1983.
[18] Johnson, L. G. and S. M. S. Jalaleddine, "MOS Implementation of Winner-Take-All Network with Application to Content-Addressable Memory," *Electronic Letters*, Vol. 27, No. 11, pp. 957-958, May 1991.
[19] Lazzora, J., et al., "Winner-Take-All Networks of O(N) Complexity," *Advances in Neural Information Processing Systems*, Touretzky D. S. (Ed.), Morgan Kaufmann, San Mateo, CA, pp. 703-711, 1989.
[20] Çilingiroglu, U., "A Charge-Based Neural Hamming Classifier," *IEEE J. Solid-State Circuits*, Vol. 28, No. 1, pp. 59-67, Jan. 1993.
[21] Grant, D., J. Taylor, and P. Houselander, "Design, Implementation and Evaluation of a High-Speed Integrated Hamming Neural Classifier," *IEEE J. Solid-State Circuits*, Vol. 29, No. 9, pp. 1154-1157, Sept. 1994.
[22] Fang, Wai-Chai, et al., "A VLSI Neural Processor for Image Data Compression Using Self-Organization Networks," *IEEE Trans.. Neural Networks*, Vol. 3, No. 3, pp. 506-518, May 1992.
[23] Robinson, M. E., H. Yoneda, and E. Sanchez-Sinencio, "A Modular CMOS Design of a Hamming Network," *IEEE Tran. Neural Networks*, Vol. 3, No. 3, pp. 444-456, May 1992.

A NEW LEARNING CONTROL LAW WITH MONOTONIC DECAY OF THE TRACKING ERROR NORM

Hyun Sang Jang and Richard W. Longman
Columbia University, New York, NY 10027

ABSTRACT

A new discrete-time learning control algorithm is presented to adjust the input to a feedback control system so that the output converges to zero tracking error during repeated executions of the command. By contrast to the most basic form of learning control, using integral control concepts in the repetition domain, this new algorithm used a system model to insure good transient behavior during the learning process, giving monotonic decay of the L_2 norm of the error history. To set the gains, the algorithm uses the system unit pulse response obtained using OKID during the learning process. The learning control action at a time step is chosen to minimize the norm of all future errors. The use of data from future time steps in the previous repetition produces a smoothing effect that results in convergence to a lower error level in the presence of noise. The learning control procedure is designed to be an automated one, that does not require prior knowledge of the system and does not require sophistication on the part of the user. The advantages of the method are illustrated in numerical examples using a model of a Robotics Research Robot.

INTRODUCTION

Learning control refers to methods of improving the tracking accuracy of a control system by learning from previous experience executing a specific command. The learning control adjusts the command given the feedback control system in order to converge on the command that produces zero tracking error. Simple and effective learning controllers can be designed requiring very little knowledge about the system, but as might be expected, additional knowledge can be used to obtain better performance, as is done here. The literature in learning control is quite substantial. The authors have contributed over 30 publications in the field that lay a rather comprehensive framework. These contributions are reviewed through 1992 in [1].

The most basic form of learning control uses the equivalent of an integral controller operating in the repetition domain at each time step. According to Arimoto's brief history of learning control [2], it started in 1984 on several continents, motivated by robots doing repetitive operations. With a precursor of [3], similar concepts appeared independently in [4] through [6], which appeared in 1984, and also in [7] by one of the authors and co-workers, which was submitted in 1984. One can show that such integral control based learning can have a very broad range of convergence, and its convergence is totally independent of the system dynamics in the discrete time domain [8], depending only on the product of the input and output matrices. However, such a learning controller can exhibit very bad transients. Certain methods of controlling these transients are discussed in [9]. The main purpose of the present work is to address this problem, guaranteeing that the error decays monotonically.

References [10-12] develop an understanding of learning control from a frequency response point of view, and develop algorithms that improve the RMS tracking error in experiments on a commercial robot by factors of 100 to 1000 in a small number of repetitions. Some of the early work in the field, e.g. [5], also takes a frequency response approach. This approach gives understanding of the source of the poor transients, and [10,11] make use of zero-phase filtering to eliminate these poor transients, but zero tracking error is sacrificed. Reference [12] eliminates this last difficulty, and presents an algorithm based on frequency response, that guarantees monotonic convergence of each frequency component of the error. The algorithm developed here produces monotonic convergence, but this time expressed in the time domain by the L_2 norm of the error history for each repetition. This avoids certain difficulties associated with using frequency response on finite time systems.

The integral control based learning control makes all of the adjustment of the control signal at a given time step based on the error measured during the previous repetition, at the time step this control choice immediately influences. This dependence on the error measured at

one time step only, makes the algorithm more susceptible to measurement noise than an algorithm that includes more time steps and gains. Also, the system input-output gain for one time step, i.e. the first Markov parameter, can be a very small number, especially when the sample time is short. These effects suggest that it could be better to develop learning control laws that use errors at more than one time step to determine the learning control action. The present paper represents one way to accomplish this.

Reference [8] presents a generalized formalism for linear learning control, and gives the stability condition that must be satisfied for convergence to zero tracking error. The general linear learning control law allows the learning signal at a given time step to be a linear combination of all errors encountered in the previous repetition. If one limits the dependence on errors at or previous to the current time step plus the delay in the system, then it is very simple to test for satisfaction of the stability condition. Various ways of choosing these gains to improve the transient performance are presented in [9]. However, in both [8] and [9] it was unclear how to benefit from learning gains multiplying errors in the previous repetition that were future to the current time step plus delay. Furthermore, when both future and past learning gains are used, the large size of the matrix whose eigenvalues determine stability can make it difficult to establish that the learning process is stable. The present work gives a solution to these issues. It supplies a methodology for setting these future gains, and the methodology independently guarantees satisfaction of the learning convergence criterion.

THE REPETITION DOMAIN FORMULATION

We consider a general discrete multivariable linear time-invariant system

$$\begin{aligned} x(k+1) &= Ax(k)+Bu(k)+w(k), \quad k=0,1,\cdots,p-1 \\ y(k) &= Cx(k), \quad k=0,1,\cdots,p \end{aligned} \tag{1}$$

The results can be generalized to linear time-varying systems. Here x is the n-dimensional state vector, y is the output and w is a forcing function including disturbances. Nominally, the matrix A is a closed-loop system matrix. For simplicity, we will use a single-input, single-output system here, although this is easily generalized. Then w contains the command to the control system, and u is the learning control signal that is added to the command to adjust the output. A repetitive process of p time steps is considered that starts from the same initial condition $x(0)$ each repetition. The disturbances in w are also considered to be repetitive, re-occurring each time the same command is executed. Following Phan and Longman [11], we can rewrite (1) in the form

$$\begin{aligned} \delta_j y(k) &= CA^k \delta_j x(0) + \sum_{i=0}^{k-1} CA^{k-i-1} B \delta_j u(i) + \sum_{i=0}^{k-1} CA^{k-i-1} \delta_j w(i) \\ \delta_j y(k) &= C\delta_j x(k), \quad k=1,2,\cdots,p \;\; ; \;\; j=1,2,3,\cdots \end{aligned} \tag{2}$$

where the backward difference operator $\delta_j z(k) = z_j(k) - z_{j-1}(k)$ represents the difference of any variable $z(k)$ in repetitions j and $j-1$. Since we are considering a process with the same initial conditions each repetition, and with a repetitive disturbance and command, $\delta_j x(0) = 0$ and $\delta_j w(i) = 0$. Writing the result for each of the p steps of repetition j, and packaging it in matrix form produces the repetition domain equations

$$\delta_j \underline{y} = P\delta_j \underline{u} \quad \text{or} \quad \underline{y}_j = I\underline{y}_{j-1} + P\delta_j \underline{u} \tag{3}$$

where an underbar indicates the matrix of the history of a variable appropriate for a repetition, and

$$\delta_j \underline{y} = \left[\delta_j y(1) \quad \delta_j y(2) \quad \cdots \quad \delta_j y(p)\right]^T \quad ; \quad \delta_j \underline{u} = \left[\delta_j u(0) \quad \delta_j u(1) \quad \cdots \quad \delta_j u(p-1)\right]^T$$

$$P = \begin{bmatrix} C(1)B(0) & 0 & \cdots & 0 \\ C(2)A(1)B(0) & C(2)B(1) & \cdots & 0 \\ C(3)A(2)A(1)B(0) & C(3)A(2)B(1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C(p)A(p-1)\cdots A(1)B(0) & \cdots & \cdots & C(p)B(p-1) \end{bmatrix}$$

Equation (3) is the needed repetition domain representation. The second version in (3) can be thought of as a state space equation in repetitions, with the system matrix being the identity matrix, and the input is the change in the learning control signal from the previous repetition. Some basic advantages of this repetition domain representation are: The repetitive disturbances are eliminated, time-varying systems become invariant in the repetition domain, and the computation of the learning control action can be made based on all of the information of the previous repetition and therefore need not be restricted to operations satisfying causality in the time domain. Note that for generality the P above is given for the time-varying case.

THE LEARNING CONTROL LAW AND THE CONDITION FOR MONOTONIC CONVERGENCE IN THE SENSE OF THE L_2 NORM

A general linear learning control law can be written as

$$\underline{u}_j = \underline{u}_{j-1} + L\underline{e}_{j-1} \quad , \quad \underline{e}_{j-1} = \underline{y}^* - \underline{y}_{j-1} \tag{4}$$

where L is the leaning gain matrix, $\underline{y}^*$ the desired feasible trajectory, and $\underline{e}_j$ is the matrix of the trajectory error history at the jth repetition. Substitution into the second of equations (3) produces an equation for the propagation of error histories from one repetition to the next

$$\underline{e}_j = (I - PL)\underline{e}_{j-1} \tag{5}$$

The condition for convergence of the tracking errors to zero for all time steps as the repetitions progress is that the matrix $(I - PL)$ have all eigenvalues less than one in magnitude, $|\lambda_i(I - PL)| < 1$, $i = 1,2,\cdots,p$ [8]. However, in spite of satisfaction of this condition, when using the simplest form of learning control in which the learning gain matrix is simply a constant scalar gain times the identity matrix, the system can have poorly behaved transients that grow to large numbers before converging to zero error, especially in the latter part of the trajectory. This is discussed in [9] from a time domain viewpoint and in [11,12] from a frequency response viewpoint.

A major aim of this paper is to develop a learning controller that has monotonic convergence of the L_2 norm of the error histories for each successive repetition. This monotonic convergence condition is expressed mathematically as $\| (I - PL)\underline{e} \|_2 < \| \underline{e} \|_2$ for all $\underline{e} \neq \underline{0}$, where $\|x\|_2 = \sqrt{x^T x}$ is the L_2 or Euclidean norm. This can be restated as

$$\|(I - PL)\underline{e}\|_2 / \|\underline{e}\|_2 < 1 \ \forall \ \underline{e} \neq \underline{0} \ ; \quad \text{i.e.,} \quad \|I - PL\|_s < 1 \tag{6}$$

where $\| A \|_s = \max_{x \neq 0} \|Ax\|_2 / \|x\|_2$ is the spectral norm. This is a stronger requirement than the convergence condition that asked the eigenvalues of $(I - PL)$ be less than one in magnitude.

Experience indicates that it is not likely that one can satisfy condition (6) with the simple integral control based learning control of a scalar gain times the identity matrix, no matter what gain is picked. An example is given in the numerical simulation section below. In the next section we develop a learning control law satisfying (6), allowing fast convergence, which is based on the nominal system model.

A NEW OPTIMAL LEARNING CONTROL LAW WITH MONOTONIC CONVERGENCE

One possible learning controller design is to pick $L = P^{-1}$ which causes the spectral norm in (6) to be zero. This requires knowledge of the system and the inverse of a large matrix. In [13] this was accomplished with on-line identification in terms of ARMA models. The inversion process can be ill-conditioned, and since the diagonal terms are often quite small for a short sample time, it can require large learning gains. Viewed in different terms, the process involved is related to inverting a discrete dynamic system. Such a process can create difficulties from imprecise cancellation of non-minimum phase zeros which are introduced by the discretization process whenever the pole excess is greater than 2 [14]. An inversion that eliminates most such difficulties is [12] where the inversion emphasizes steady state behavior. Here we develop another alternative for producing monotonic convergence that does not require an inversion, and satisfies a chosen optimality condition.

There is considerable freedom in the choice of L. Consider using this freedom to optimize a cost criterion making the choice of control action for time step k such that it minimizes the sum of the squares of the future tracking errors

$$f(u_j(k)) = \tfrac{1}{2}\left[e_j^2(k+1) + e_j^2(k+2) + \cdots + e_j^2(p)\right] = \tfrac{1}{2} E_j^T(k) E_j(k) \tag{7}$$
$$E_j(k) = \left[e_j(k+1) \quad e_j(k+2) \quad \cdots \quad e_j(k+p)\right]^T$$

assuming no change in the learning control signals for future time. It is clear that if one minimizes this function at every time step, then one has arrived at zero error. Consider the use of the steepest descent method to minimize this function as the repetitions progress

$$u_{j+1}(k) = u_j(k) + \Delta u_j(k) = u_j(k) - s[\partial f / \partial u_j] = u_j(k) - s[\partial E_j / \partial u_j] E_j \tag{8}$$

where $[\partial f / \partial u_j]$ is the column matrix of gradient components, the square bracket in the last expression has the components defined appropriately, and s is a scalar chosen to influence the speed of convergence. In the time-varying case

$$\partial E_j / \partial u_j = -\left[C(k+1)B(k) \quad C(K+2)A(K+1)B(k) \quad \cdots \quad C(P)A(p-1)\cdots A(1)B(k)\right]$$

and otherwise from (2)

$$\partial E_j / \partial u_j = -\left[CB \quad CAB \quad CA^2B \quad \cdots \quad CA^{p-1}B\right]$$

Substituting in (8) we obtain the desired learning control law

$$u_j(k) = u_{j-1}(k) + s \cdot \sum_{i=1}^{p} CA^{i-1} B e(k+i) \quad ; \quad k = 0,1,\cdots,p-1 \tag{9}$$
$$e(k+i) = 0, \quad k+i > p \quad ; \quad j = 1,2,3,\cdots$$

which corresponds to the choice of the learning control gain matrix as

$$L = sP^T \tag{10}$$

For sufficiently small positive values of s this learning control law satisfies condition (6) for monotonic convergence. With this law the spectral norm $\|I - PL\|_s$ becomes the norm of a symmetric matrix so that

$$\left\|I - sPP^T\right\|_s = \max_i \left|\lambda_i\left(I - sPP^T\right)\right| = \max_i \left|I - s\lambda_i\left(PP^T\right)\right| < 1$$

Rewriting the inequality as $-1 < 1 - s\lambda_i(PP^T) < 1$ for all i, produces the range of values of s

$$0 < s < 2/\| P \|_s^2 \tag{11}$$

for which the learning process has the desired property that it produces monotonic decay of the Euclidean norm of the tracking error history for each repetition as the repetitions progress.

ESTIMATION OF LEARNING GAINS FROM MARKOV PARAMETER IDENTIFICATION

The algorithm developed above requires that one know the system Markov parameters for use as learning control gains. These can be obtained from general input-output data by using the OKID (Observer Kalman Filter Identification) algorithm which operates in the time domain [15]. The algorithm makes use of an observer, finding first the Markov parameters of the observer, and from these it finds the Markov parameters of the system. If desired one can use relatively short data records, and use the algorithm to construct a longer Markov parameter record. In place of OKID, one can use standard fast Fourier transform techniques, which obtain the parameters as the inverse FFT of the ratio of the FFTs of the output and input signals. Note that these methods apply to time-invariant systems, and an alternative must be used in the case of time-varying systems. Provided $\|I - PL\|_s$ remains less than 1, estimation errors can be tolerated and still maintain the monotonic convergence property. This expresses the degree of robustness of the proposed learning algorithm based only on an initial model. The approach proposed here, parallels that in [12], and asks for re-identification of the Markov parameters from the data at any repetition for which the error norm increases. If the model is right, monotonic convergence must occur. If the error increased at a repetition, the model is incorrect and the data for that repetition should be rich in the information needed to correct the model. Thus, by applying OKID in this way the algorithm becomes self correcting to initial errors in the model.

A GENERALIZATION

Suppose that we generalize the learning control to $L = SP^T$ which includes a gain matrix S. Denote the singular value decomposition of P as $P = Udiag(\sigma_1, \sigma_2, ..., \sigma_p)V^T$ and choose the gain matrix as

$$S = Vdiag(s_1, s_1, ..., s_1)V^T \tag{12}$$

With this substitution we obtain $I - PL = U(I - diag(s_1\sigma_1^2, s_2\sigma_2^2, ..., s_p\sigma_p^2)U^T$, from which it is obvious how to adjust the scalars $s_1, s_1, ..., s_1$ in order to influence not only the spectral norm in (6) but also to influence the speed of convergence of each error component from projection on the orthonormal columns of U.

NUMERICAL SIMULATION OF A ROBOT APPLICATION

In order to demonstrate the advantage of the proposed algorithm over the basic form of learning control using integral control applied in the repetition domain, we make a numerical simulation study using to a model of the Robotics Research Cooperation K-series 807iHP manipulator. This manipulator has been studied in [10], where it is shown that a rather good model of the manipulator is as a set of 7 uncoupled time-invariant third-order systems, consisting of the same transfer function for each link

$$G(s) = [\alpha/(s+\alpha)][\omega_n^2/(s^2 + 2\zeta\omega_n s + \omega_n^2)] \tag{13}$$

where α = 8.8 rad/s (1.4 Hz), ζ = 0.5, and ω_n = 37 rad/s (5.9 Hz). The decoupling is due to the high gear ratios used, that produce large diagonal elements in the inertia matrix. The real pole is introduced by the controller to obtain low pass characteristics limiting the excitation of the first resonant mode of the system, and the complex conjugate pair represents a resonance. We pick a polynomial desired trajectory of the form

$$y_d = 5\pi \cdot t^3 - 7.5\pi \cdot t^4 + 3\pi \cdot t^5, \qquad 0 \le t \le 1 \tag{14}$$

White measurement noise is added in some of the simulation runs, with a noise level equal to 1% of the maximum of the desired output (1.6 radians).

Figure 1 shows the robot trajectory obtained by commanding this trajectory to the feedback controller. Substantial tracking error is observed. The most basic form of learning control applies integral control concepts in the repetition domain, e.g. as discussed in [7,8], and can be expressed mathematically in the present context by setting the learning control matrix L of (4) equal to a constant times the identity matrix. We pick the constant to be unity, which is the natural value of the learning control gain in this system as defined in [10]. Figure 2 shows the behavior of the Euclidean norm of the error history as a function of the repetitions when this learning control law is applied to modify the command to the feedback system. The error decreases dramatically, by several orders of magnitude over the first few repetitions. Then the error begins to grow. This does not indicate that the learning process is unstable -- it is guaranteed that the process will converge to zero tracking error. Nevertheless, the error grows dramatically, building up the amplitude of originally small error components at frequencies for which the phase lag in the closed loop system is around 180 deg [10-12]. Figure 3 shows the corresponding error histories for repetitions 10, 20, and 30. Obviously, the convergence does not exhibit the desired property of monotonic convergence of the tracking error norm. One might ask, is there a learning gain other than the 1 used here, that would produce monotonic decay of the error? Figure 4 shows that the answer is no, that for all positive learning gains the spectral norm inequality in (6) is violated.

In practice, the temporary build up of error at high frequency in Fig. 3 is unacceptable. In this example, one could simply stop the learning at roughly repetition 8 after obtaining two orders of magnitude improvement in tracking accuracy, but in other systems one might not obtain sufficient improvement before the poor transient behavior develops. The algorithm of this paper allows one to obtain further improvement with a monotonic decay of the error norm, as illustrated in Figs. 5 and 6. These figures use the learning control law (10) with the value of s set to 1, and with the $CA^{i-1}B$ computed using OKID. In these simulations OKID is used between every repetition, recomputing the learning control gains every repetition. In practice better results might be obtained by using reidentification less often as discussed below. Figure 7 shows the deviations of the estimated values of the Markov parameters from their true values at repetition 30. Figure 5 shows that in the presence of noise, the learning control algorithm proposed here can eliminate essentially all systematic error, allowing one to produce final tracking errors near the noise level of the system.

The matrix P is of large dimension, creating a large number of learning control gains in (10), and a substantial number of multiplications to perform between repetitions in order to compute the learning control action for the next repetition. In a stable system, the Markov parameters decay with time, making it unnecessary to retain the full P^T matrix in (10). In practice, one can set $CA^kB = 0$ for all k above a certain threshold, p. Figure 8 shows the error norm as in Fig. 5, but using a threshold of 10, 20, 30 and 100. Threshold values of 10 and 20 produce monotonic decay, but at a slower rate. Threshold values of 30 and above produce essentially indistinguishable results with fast convergence.

We conclude that the proposed learning control algorithm gives much better performance than integral control based learning, by producing monotonic decay of the transients, and by producing less sensitivity to noise which allows a lower final RMS error in the presence of noise.

DISCUSSION AND CONCLUSIONS

In conclusion we make the following comments regarding the new learning control algorithm developed in this paper:

1. The algorithm is intended as an automated procedure for producing large improvements in the tracking accuracy of a feedback control system executing a repetitive task, without requiring knowledge and expertise on the part of the user.

2. The algorithm is designed to produce monotonic decay of the Euclidean norm of the tracking errors as the repetitions progress, i.e., it insures well behaved transients during the learning process. For many systems, the basic form of learning control based on integral control concepts is unable to produce such monotonic convergence no matter what value of learning gain is chosen, which can be understood from the frequency domain analysis in

[10,11]. The learning control temporarily amplifies frequency components for which the feedback control has a 180 deg phase lag. In [10], a noncausal zero-phase low pass filter is used with integral control based learning to ignor the part of the error violating the monotonic convergence. In [11], compensation is introduced allowing one to increase the bandwidth of the low pass filter, but zero tracking error is sacrificed in bothe cases. Reference [12] is the most similar to the present work in terms of objective and accomplishment. It produces monotonic decay of the error in terms of its steady state frequency response components. However, the algorithm produced here has the advantage that it requires less computation and is simpler, does not require transformation of data to the frequency domain, and is not subject to approximations needed in [12] to apply frequency response methods to a finite time signal.

3. The learning control law developed satisfies the optimality condition that the change in the control action for any given time step from one repetition to the next is such that it minimizes the norm of all future errors based on a system model. This can produce smoother behavior than the one step ahead nature of most learning control algorithms. Since errors at all future steps, rather than just the next step, influence the choice of learning control action, there is significant smoothing of noise. This is helpful during convergence, but it is perhaps more significant when the learning control approaches the limit of learning based on the noise level in the system. Because of this smoothing, one can expect that the final RMS tracking error level, responding to a steady state noise level, will be much better for the present algorithm.

4. In order to obtain these desirable properties, the algorithm needs to know the system expressed in terms of the unit pulse response history, i.e. the Markov parameters. The OKID algorithm is designed for this purpose. It is easier to obtain accurate values of the Markov parameters than it is to obtain a system model. Certain decisions such as the proper system order to use are avoided, and less error occurs when there does not exist such a linear differential equation model, for example in the presence of mild nonlinearities.

5. The learning gains are based on a model, and the monotonicity of the result will be maintained provided that the condition (6) is satisfied using the true P of the system in (6), but the P of the model in the learning law. It is expected that this condition gives a wide robustness region. Nevertheless, if the error norm fails to decay at some repetition, then data is generated that is accentuated in that part of the system that needs to be known better in order to satisfy (6). In the simulations here, we have used OKID every repetition, but experimental experience in [12] on a similar problem suggests that better results are obtained if reidentification is used more sparingly. System identification is not an altogether straightforward field. Reidentification every repetition can fail to allow signals to grow out of the noise level. On the other hand, using OKID on a signal after significant growth of the error uses data after much learning has already been accomplished. This may emphasize only parts of the model that were not well modeled by an earlier application, and because of this emphasis in the data, the parts known well by previous identification may then become inaccurate. One can include both early and later data in the OKID identification to handling this. The compromise used in [12] makes reidentifications only at the a repetition violating the monotonic decay property, using the data obtained at that repetition.

6. It is sometimes stated that since one has an identified model, one could simply solve it for the needed input to produce the desired output, and would not need learning control to produce zero tracking error. Reference [10] shows that this is false in practice, that one can use this solution as the starting point for learning control and it will produce significantly further improvement of the tracking errors.

7. The algorithm has a large number of learning gains. Since the computations are being done between repetitions this may not be a serious difficulty. In addition, since learning control is nominally applied on a feedback control system which would automatically be asymptotically stable, the Markov parameters decay with sample time. Thus, one can limit the number used to the number of time steps in a few time constants of the system.

8. A generalization of the algorithm is presented which allows one to adjust the learning rate for each component of the error history as projected onto a specific orthonormal basis. A singular value decomposition of matrix P is required to obtain the basis. One should not be too aggressive in adjusting the convergence rate since doing so approaches inverting the system.

9. In [8] a general formulation for linear learning control was presented, allowing the use of data from the previous repetiton that is future to the time of the learning signal being computed. No systematic method of setting such future gains was available. The approach

used here picks such gains, establishes stabiltiy of the learning process, and obtains the monotonic and smoothness advantages discussed above.

REFERENCES

1. R. W. Longman, M. Phan, and J.-N. Juang, "An Overview of a Sequence of Research Developments in Learning and Repetitive Control," *Proceedings of the First International Conference on Motion and Vibration Control (MOVIC)*, Yokohama, Japan, September 1992.
2. R. W. Longman, Editor, *Supplementary Material, Tutorial on Learning Control*, Organizers: R. W. Longman, S. Arimoto, Additional Lecturers: M. Phan, and J.-N. Juang, 1991 IEEE Conference on Intelligent Control, Arlington, VA, August 1991. Contains: Arimoto, S., "Introduction: Brief History of Development of Learning Control in Robotics."
3. M. Uchiyama, "Formulation of High Speed Motion Pattern of a Mechanical Arm by Trial," (in Japanese), *Transactions of the Society for Instrumentation and Control Engineers*, Vol. 14, 1978, pp. 706-712.
4. S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering Operation of Robots by Learning," Journal of Robotic Systems, Vol. 1, No. 2, 1984, pp. 123-140.
5. J. J. Craig, "Adaptive Control of Manipulators Through Repeated Trials," *Proceedings of the American Control Conference*, San Diego, CA, 1984, pp. 1566-1573.
6. G. Casalino, G. Bartolini, "A Learning Procedure for the Control of Movements of Robotic Manipulators," *IASTED Symposium on Robotics and Automation*, Amsterdam, 1984, pp. 108-111.
7. R. H. Middleton, G. C. Goodwin and R. W. Longman, "A Method for Improving the Dynamic Accuracy of a Robot Performing a Repetitive Task," *International Journal of Robotics Research*, Vol. 8, No. 5, Oct. 1989, pp. 67-74. Also available as University of Newcastle, Newcastle, Australia, Department of Electrical Engineering Technical Report EE8546, 1985.
8. M. Phan, and R. W. Longman, "A Mathematical Theory of Learning Control for Linear Discrete Multivariable Systems," *Proceedings of the AIAA/AAS Astrodynamics Conference,* Minneapolis, Minnesota, August 1988, published by AIAA, Washington, DC, pp. 740-746.
9. C.-K. Chang, R. W. Longman, and M. Phan, "Techniques for Improving Transients in Learning Control Systems," *Advances in the Astronautical Sciences*, American Astronautical Society, Vol. 76, 1992.
10. H. Elci, M. Phan, R. W. Longman, J.-N. Juang, and R. Ugoletti, "Experiments in The Use of Learning Control for Maximum Precision Robot Trajectory Tracking," *Proceedings of the 1994 Conference on Information Sciences and Systems,* Princeton, NJ, 1994.
11. H. Elci, R. W. Longman, M. Phan, J.-N. Juang, and R. Ugoletti, "Discrete Frequency Based Learning Control for Precision Motion Control," Proceedings of the 1994 IEEE Conference on Systems, Man, and Cybernetics, San Antonio, TX, October 1994, to appear.
12. H. Elci, M. Phan, R. W. Longman, J.-N. Juang, and R. Ugoletti, "Automated Learning Control Through Model Updating for Precision Motion Control", 1994 ASME Winter Annual Meeting, Adaptive Structures and Material Systems Symposium, Nov. 1994, to appear.
13. M. Phan and R. W. Longman, "Indirect Learning Control with Guaranteed Stability," *Proceedings of the 1989 Conference on Information Sciences and Systems*, Johns Hopkins University, Baltimore, MD, 1989, pp. 125-131.
14. R. W. Longman and E. J. Solcz, "Small Gain Robustness Issues in the P-Integrator Repetitive Controller," *AIAA/AAS Astrodynamics Conference, A Collection of Technical Papers*, Part 2, Portland, Oregon, August 1990, pp. 537-551.
15. J.-N. Juang, M. Phan, L. G. Horta, and R. W. Longman, "Identification of Observer/Kalman Filter Markov Parameters: Theory and Experiments," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 2, March-April 1993, pp. 320-329.

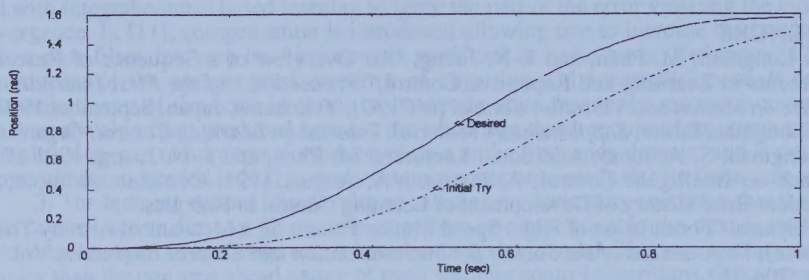


Fig. 1 Desired trajectory and the corresponding trajectory obtained with feedback control alone.

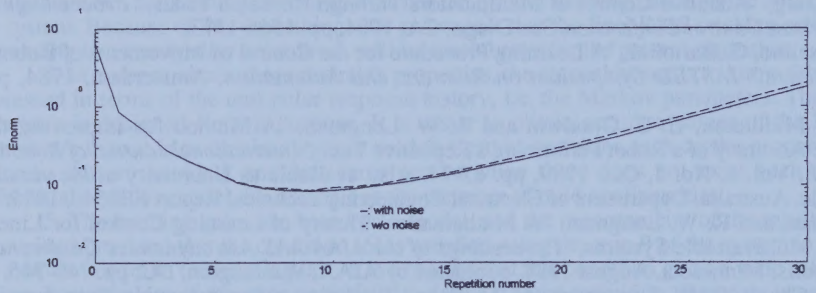


Fig. 2 L_2 norm of the errors for successive repetitions using integral control based learning.

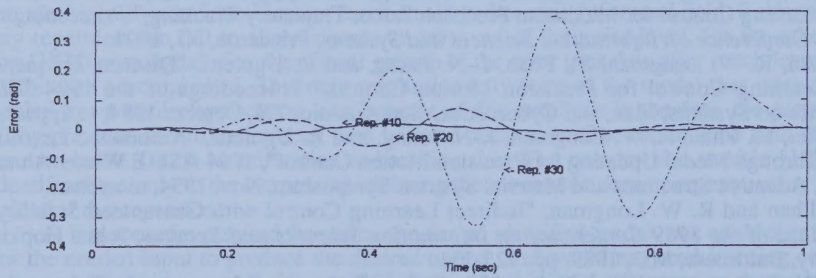


Fig. 3 Tracking error history for repetitions 10, 20, and 30 using integral control based learning.

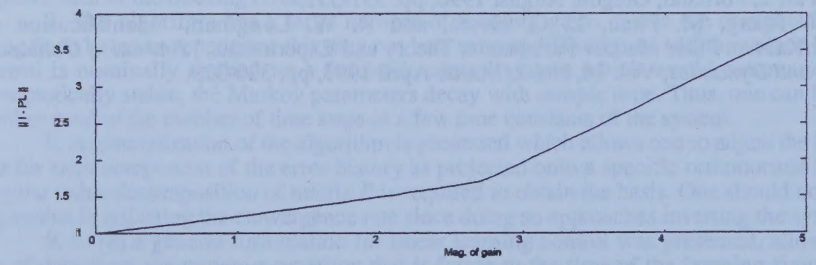


Fig. 4 The spectral norm of *I-PL* as a function of gain *s* for integral control based learning, $L = sI$.

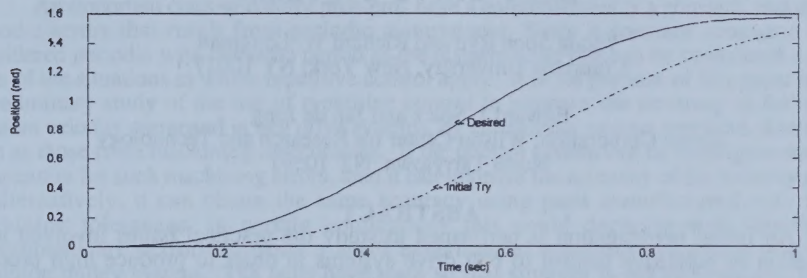


Fig. 1 Desired trajectory and the corresponding trajectory obtained with feedback control alone.

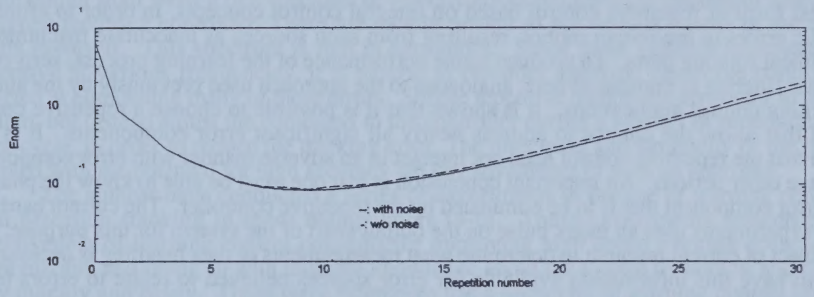


Fig. 2 L_2 norm of the errors for successive repetitions using integral control based learning.

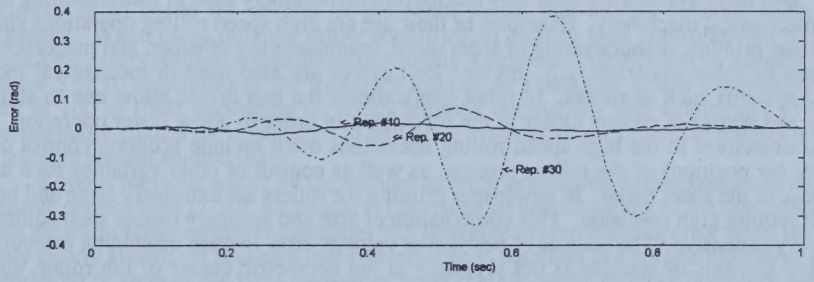


Fig. 3 Tracking error history for repetitions 10, 20, and 30 using integral control based learning.

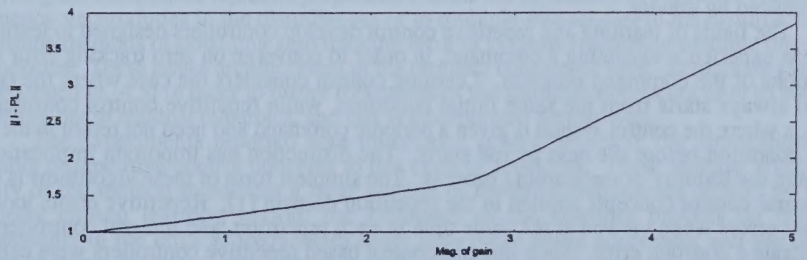


Fig. 4 The spectral norm of *I-PL* as a function of gain *s* for integral control based learning, $L = sI$.

EXPERIMENTS IN THE USE OF REPETITIVE CONTROL IN BELT DRIVE SYSTEMS

Yeong Soon Ryu and Richard W. Longman
Columbia University, New York, NY 10027

Edward J. Solcz and Jan de Jong
Xerox Corporation, Wilson Center for Research and Technology
North Tarrytown, NY 10591

ABSTRACT

An initial investigation is performed to study the practical issues involved in the application of repetitive control to belt drive systems in order to produce high precision motions. Such systems occur in many manufacturing operations and in other mechanical machinery. Some applications include newspaper printing, the manufacture of sheet metal and rolls of other materials, and belt drives in robot arms. The approach used here applies the simplest form of repetitive control based on integral control concepts, in order to eliminate periodic errors in the output motion resulting from such sources as inaccurate machining of mechanical rotating parts. To produce stable performance of the learning process, zero-phase low-pass filtering is introduced here, analogous to the approach used previously by the authors in learning control applications. It is shown that it is possible to choose a repetitive control period that allow the control to address nearly all significant error components. It is also shown that the repetitive control need not interact in an adverse manner with error components that have other periods. An important conclusion is that one must be able to know the phase of each error component that is to be eliminated by the repetitive controller. The current hardware in the experiments uses an index pulse on the output shaft of the system for this purpose. It is the subject of current research to determine what measurements or data handling is necessary in order to have this information available for error sources believed to relate to errors in the machining is bearing parts.

INTRODUCTION

Belt drive systems are used very extensively in machinery used in manufacturing and in other mechanical machinery. Examples of their use are high speed rolling operations such as newspaper printing, manufacturing of sheet metal, manufacturing of paper, and manufacturing of sheets of other materials, e.g., mylar. Belt drives are also used in mechanical motion control systems such as robots. In robot applications, the belt drives allow one to shift the weight and inertia of motors closer to the robot base in order to allow faster operation. The control objectives in the high speed rolling operations often include accurate control of the velocity (or position) of the rolling process, as well as control of other variables such as the thickness of the sheet metal. In newspaper printing the rollers are extremely large and heavy, and yet require high precision. This combination of size and accuracy causes such equipment to be very expensive. The sources of position or velocity error include machining inaccuracies in which the axis of rotation is not precisely at the geometric center of the roller, similar inaccuracies in the machining of the roller bearings and associated elements, flexibility of the belts used, and in some cases, slippage of the belts. They are also subject to inaccuracies from sag produced by gravity.

The fields of learning and repetitive control develop controllers designed to learn from previous experience executing a command, in order to converge on zero tracking error as the repetitions of the command progress. Learning control considers the case where the control system always starts from the same initial condition, while repetitive control considers the situation where the control system is given a periodic command and need not return to the same initial condition before the next period starts. The distinction has important implications in assessing the stability of the learning process. The simplest form of these algorithms is based on integral control concepts applied in the repetition domain [1]. Repetitive errors look like constant errors when viewed at the same time in each repetition, and integral controllers will not tolerate a constant error. Such integral control based repetitive controllers were called *p*-integrators in [1], which is one of the first works in the field. Reference [2] is a later

publication presenting similar ideas. In [3], a modern control formulation of such concepts is generated for the learning control problem.

An important class of control problems have a command that is a constant, and there are periodic errors that result from periodic disturbances. Since a constant command can be considered periodic with the same period as the disturbances, this can be considered a special class of the situations to which repetitive control apply. It is the purpose of this paper to make a preliminary study of the use of repetitive control to improve the accuracy in following a constant velocity command in belt drive systems, by eliminating various repetitive disturbances such as those from machining inaccuracies. If the control system can be intelligent enough to compensate for such machining errors, then it can improve the accuracy of the velocity control, or alternatively, it can obtain the same accuracy using parts manufactured with relaxed machining tolerances. In certain applications this could decrease cost significantly. Experiments are performed using a small experimental apparatus consisting of a double reduction pulley system. The belts have teeth so that slippage is not an issue, but they are flexible. Thus, the physical system consists of three rolling shafts having inertia, connected successively by two belts that have flexibility. Fast Fourier transforms applied to the velocity of the output shaft show a rich spectrum of velocity errors remaining in the output using a well designed feedback controller. These are mostly related to the various speeds of the rotating shafts involved. This spectrum is rich in harmonics. Since these velocity inaccuracies are repetitive at various frequencies, it is natural to consider applying repetitive control ideas targeted at eliminating many of these errors. Simple repetitive controllers do not try to learn the system, but instead adjust the command to the controller during a number of repetitions of the error cycle, in order to reduce the tracking error. The directness and simplicity of the methods coupled with the potential for high accuracy are key advantages of such an approach.

The authors have conducted a series of experiments in the use of learning control for high precision tracking in robot manipulators [4-6]. These experiments demonstrate the effectiveness of learning control concepts for improving high speed tracking accuracy by a large margin, and doing so with minimal knowledge about the system dynamics, and doing so very quickly and easily. These three references use a hierarchy of learning control laws, and study the ability of these laws to reduce errors in a large angle trajectory involving all seven joints of a redundant 7 degree-of-freedom Robotics Research commercial robot manipulator. In order to maximize the difficulty and nonlinearity of the problem, this trajectory reaches the maximum rated speed for several of the robot joints, and travels through much of the workspace. When using only the rather sophisticated built-in joint controllers on this robot, the tracking errors reached a maximum of roughly 9 degrees. Application of learning control laws for 5 or 10 repetitions of the trajectory resulted in a decrease in this tracking error by a factor of 100 to 1000, depending on which learning control law was employed. This paper experimentally studies the application to belt drive systems, of the repetitive control analog of the learning control studied in [4].

Several previous works by the authors develop approaches to designing and analyzing repetitive control laws [7-10]. Reference [7] develops an understanding of the conditions under which the *p*-integrator repetitive controller will converge to zero tracking error when the system is only poorly known. It is shown that convergence is obtained in a surprisingly large number of situations in spite of the large number of poles that lie on the stability boundary before the learning gain is increased from zero. References [8-10] develop the modeling needed for analyzing the disturbance rejection properties, discuss the robustness of repetitive controllers to frequency uncertainties in the disturbances or to multiple frequencies in the disturbances, and determine methods of obtaining zero tracking error in systems containing multiple disturbance frequencies. These issues are important in the problem considered here. Zero phase filtering is incorporated in [4,5] in order to insure robustness of the learning process not only to inaccurate modeling but to residual dynamics, and this is incorporated here as well.

The experiments reported here investigate several issues in the application of the learning or repetitive control concepts to this class of problems. These include: studying the ability of the method to address a large portion of the error frequencies present by using a properly chosen learning period; the influence of the learning process on frequency components in the error at frequencies other than those being addressed; and the practical issues of using noncausal zero-phase low-pass filtering in the repetitive control problem in order to aid in insuring stability of the learning process.

REPETITIVE CONTROL BASED ON INTEGRAL CONTROL CONCEPTS

References [1,7] develop the simplest form of learning control based on integral control concepts applied in the repetition domain. Suppose that the period of the input and repetitive disturbance is p time steps ([11] discusses the influence of having the disturbance be a noninteger number of time steps). Then the repetitive control signal to be added to the command input of the feedback control system at time step k' of repetition $j+1$ of the period is

$$u_{j+1}(k') = \Phi \sum_{i=1}^{j} e_i(k'+1) \tag{1}$$

The subscript indicates the repetition number, which starts at 0 for the first period in which feedback control only is applied, and then with the second period, or repetition 1, the learning control adjustments begin. The Φ is the learning gain. This formula assumes that there is a one time step delay between a change in the feedback control system input and the sample time in which the corresponding change in the output is first observed. This produces the one step ahead used in the errors $e_i(k'+1)$. The summation represents the analog of an integral applied in the necessarily discrete repetition domain.

For simplicity of computation a recursive form can be used. Note that the true time step k corresponding to step k' of repetition j is given by $k = jp + k'$. Then dropping the subscript and converting the argument to the true time step relative to the starting time gives the repetitive control law

$$\begin{aligned} &u(k+p) = u(k) + \Phi e(k+1) \\ &k = 1,2,3,... \end{aligned} \tag{2}$$

The experiments reported in this paper make use of this p-integrator repetitive control law. However, there are certain modifications relating to periodic restarting of the system, together with zero-phase low-pass filtering of the repetitive control between restarts as discussed below. A natural choice for the initial conditions is

$$u(k) = y^*(k+1) \tag{3}$$

for all steps k in the first period, where $y^*(k)$ is the desired output at time step k. This anticipates the one step delay of the feedback system in responding to the command at step k. Taking the z-transfer function of (2), and using zero initial conditions produces its transfer function

$$U(z) = \frac{\Phi z}{z^p - 1} E(z) \tag{4}$$

In order to analyze the effect of this repetitive control on the tracking error of the system, consider for simplicity the case of applying the control to the input of a continuous time feedback control loop (the way in which the mathematics is altered in the case of a feedback loop using a digital controller can be found in [8]). Let the closed loop transfer function of the feedback control system be $G(s)$, and compute the equivalent z-transfer function associated with using a zero order hold on the input computed from (2)

$$G(z) = (1 - z^{-1}) Z\left[\frac{G(s)}{s}\right] = \frac{N(z)}{D(z)} \tag{5}$$

where $N(z)$ and $D(z)$ represent the numerator and denominator polynomials of $G(z)$. Consider first the case of a periodic command with no periodic disturbance. Then the error history satisfies

$$E(z) = Y^*(z) - Y(z) = Y^*(z) - G(z)U(z) = Y^*(z) - \frac{\Phi z N(z)}{(z^p - 1)D(z)} E(z) \qquad (6)$$

or equivalently

$$\left[(z^p - 1)D(z) + \Phi z N(z)\right]E(z) = D(z)(z^p - 1)Y^*(z) \qquad (7)$$

This can be thought of as a difference equation by recognizing that $z^i E(z)$ in the sample domain is a shift of i steps. Since the desired trajectory $Y^*(k)$ is periodic, $(z^p - 1)Y^*(z)$ gives an identically zero forcing function to the difference equation. Hence, (7) is a homogeneous difference equation whose solution is the error history at the sample times. The general solution can be written as a sum of the $n+p$ linearly independent solutions (where n is the number of poles of $G(s)$)

$$e(k) = c_1\rho_1^k + c_2\rho_2^k + \ldots + c_{n+p}\rho_{n+p}^k \qquad (8)$$

where c_i are the arbitrary constants determined by the initial conditions, and the ρ_i are the $n+p$ roots of the characteristic equation

$$(z^p - 1)D(z) + \Phi z N(z) = 0 \qquad (9)$$

The form used in (8) assumes all of the roots of (9) are distinct, otherwise terms of the form $k^j\rho_i^k$ appear. Convergence of the error $e(k)$ to zero as k gets large requires that $|\rho_i| < 1$ for all i, and the convergence rate is determined by how much smaller than one the magnitudes of these roots are.

The influence of error sources such as inaccurate machining of the shafts in the belt drive system is actually a nonlinear influence based on geometrical relations. It is reasonable to approximate its influence by considering an additive disturbance $V(z)$ to the output of the feedback control system $Y(z)$. Then (6) becomes $E(z) = Y^*(z) - G(z)U(z) - V(z)$, and the difference equation (7) again applies provided that $Y^*(z)$ is replaced by $Y^*(z) - V(z)$. Hence, any disturbances that are periodic with period p time steps are also eliminated, and disturbances not having this period produce a nonzero forcing function to the difference equation, producing a nonzero error. The error produced by this class of disturbances will be studied here.

THE EXPERIMENTAL SETUP

The experimental setup used to investigate practical issues of applying repetitive control concepts on rolling operations, consists of a double reduction timing belt system. The input axis is driven by a model FN38 motor driven by an amplifier. This shaft is connected to an idler shaft by a belt. The other end of the idler shaft has a larger radius, and this side is connected to the output shaft with a second belt. The belts have teeth so that there is no slipping, but the rubber like material of the belt may allow elastic behavior. In the middle of one side of each belt is a tensioning pulley, that is spring loaded to apply force to the belt and maintain tension. The output shaft has a small flywheel load, and is connected to a series 835 rotary incremental encoder that produces 10800 pulses per revolution. This is a rather typical example of a double reduction drive system that might be used in many rolling operations.

The teeth used on the belts indicate that the nominal motion of each shaft is related as a ratio of integers. For every rotation of the output shaft the input shaft rotates through 9 revolutions, and the idler shaft through 4.5 revolutions. Slight variations in the actual rotation rates occur due to variations in the tension of the belts, and with any elastic behavior of the belt.

The motor torque applied to the input shaft is adjusted by a feedback control system, using the encoder measurements on the output shaft, in order to try to obtain constant angular velocity of this shaft. For the experiments the desired velocity was one revolution per second. The control law employed consists of two zeros and two poles

$$\alpha \frac{(s/\omega_o)^2 + 2\beta_o(s/\omega_o) + 1}{(s/\omega_p)^2 + 2\beta_p(s/\omega_p) + 1} \tag{10}$$

where α = 69782947.5, ω_o = 15.86 Hz, β_o = 0.908, ω_p = 0.01 Hz, and β_p = 0.69. This is designed to make a high gain controller at low frequencies, with a large bandwidth, producing a phase margin of 25 degrees and a 100% gain margin. Figure 1 gives experimental data for the magnitude and phase of the closed loop frequency response of this feedback control system. Note that for frequencies from roughly 10 to 100 Hz there is some amplification of the input sinusoid, and that the phase lag in the output reaches 180 degrees around 90 Hz.

The current configuration of the hardware has an index pulse each revolution of the output shaft. This can be used for synchronization of the repetitive control signal with the error sources.

THE DISTURBANCE ENVIRONMENT AND CHOICE OF REPETITIVE CONTROL PERIOD

Under steady operating conditions the velocity of the output shaft exhibits small fluctuations in velocity. Figure 2 shows the frequency spectrum of this error up through 70 Hz obtained from 16 seconds of data at a 1000 Hz sample rate. We wish to investigate the potential of repetitive control to attenuate as many of these error frequency components as possible.

The shaft ratios help to interpret the frequencies observed in this plot. The output shaft is rotating at 1 revolution per second to within the error of the control system, so that errors associated with each rotation of this shaft will have a fundamental frequency of 1 Hz. Harmonics of the once around error signature could produce all integer multiples of 1 Hz. The input shaft nominally rotates 9 times for each rotation of the output shaft, so that once around errors from this shaft produce a 9 Hz fundamental with all integer multiples of 9 Hz for the harmonics. The once around errors from the the idler shaft have frequencies of 4.5 Hz and its integer multiples. Examination of Fig. 2 shows that if one includes 4.5 Hz and all harmonics (which automatically includes the 9 Hz and its harmonics), that one obtains frequencies corresponding to approximately half of the large peaks in the error frequency spectrum. The absence of integer multiples of 1 Hz except for those that are also integer multiples of 9 Hz, suggests that once around errors in the output shaft are much less significant than those of the other two shafts. Note that errors related to tooth engagement and disengagement will appear as harmonics of the once around frequency.

The error spectrum seems to have one substantial peak in the middle between each of the harmonics of 4.5 Hz, i.e., corresponding to multiples of 2.25 Hz. Such a frequency corresponds to twice around errors for the idler shaft, and could be due to machining inaccuracies in the ball bearing assembly for the shaft.

Note that all of the frequencies cited are associated with the nominal speed of the output shaft and with once around or twice around errors. The output shaft velocity is not constant; it exhibits the error spectrum in Fig. 2. Therefore, one expects the once or twice around error peaks to exhibit a frequency modulation based on the velocity fluctuations.

In applying repetitive control concepts one must pick the period for the learning process. Using the existing hardware with an index pulse for each revolution of the output shaft, it is natural to consider using a period of 1 second corresponding to the period between index pulses. Such a repetitive controller attempts to decrease all errors having this as one of the possible periods. The errors at 1 Hz have a period of 1 second. In addition, all harmonics also have this period. For example, the errors at 9 Hz are periodic with period 1/9th of a second, but also periodic with period 1 second.

Hence, using a 1 second learning repetitive control period attempts to decrease all integer frequencies, and the large peaks that correspond to integer frequencies are 9 Hz, 18 Hz, 27 Hz, 36 Hz, ... Using a repetitive control period of 2 seconds, allows one to include the 4.5 Hz error components as well as its harmonics, since a 4.5 Hz signal has 9 periods in 2 seconds, and hence is periodic with period 2 seconds. Using a repetitive control period of 4 seconds allows one to address the 2.25 Hz and harmonic frequencies.

From physical considerations one expects that there can be other periods that are not necessarily related to the output shaft rotation through a relation of a ratio of integers. One

such source is a natural frequency related to vibrations of the belt as it interacts with the spring loading of the tensioning pulley. Generically, no choice of an integer number of seconds would allow one to learn these frequencies, although by choosing a sufficiently long period one could come arbitrarily close to matching the period. Reference [9,10,11] present some of the implications for repetitive control, and some of the methods of addressing the associated problems.

REPETITIVE CONTROL VIEWED IN TERMS OF STEADY STATE FREQUENCY RESPONSE

References [4,5,6] present a frequency response interpretation of learning control, and discuss the relationship between the true stability criterion and a criterion obtained from steady state frequency response thinking. The frequency response stability criterion is put in perspective as a criterion to use to insure well behaved transients during the learning process, but does not produce the stability boundary. Here, we apply similar thinking to the repetitive control problem.

We summarize below the thinking in heuristic terms; refer to [6, 5] for a more rigorous treatment. Suppose for the sake of illustration that the dc gain of the feedback control system is 1, we choose a repetitive control gain of 1, and we examine the changes produced from one period to the next by the repetitive controller in terms of steady state frequency response analysis, ignoring transients. Then if in the first period there were a constant error, the repetitive control law would subtract that constant from the command in the next period, and because of the dc gain of 1, this change would correct the error in the steady state.

Now consider an error at a higher frequency for which the feedback control system has a decreased output and a phase lag. If we ignore the phase lag, the effect of a slightly decreased output is that the corrective action of the repetitive controller makes a corrective adjustment of the command that is somewhat too small to completely correct the error in the next repetition. A small phase lag will have a similar effect.

For most systems, there will be a frequency for which the phase lag reaches 180 degrees. The changes made by the repetitive controller will then be shifted by 180 degrees as they go through the feedback system, giving the error adjustment the wrong sign and adding to the error instead of decreasing the error. The boundary line between when the repetitive control action decreases and increases the error in terms of steady state frequency response is roughly at a phase lag of 90 degrees. The precise value is given by the frequency at which the Nyquist plot leaves a unit circle centered at +1 (see [6] and references therein).

If we apply this steady state thinking to the feedback controller with the Bode plot in Fig. 2, we cannot allow error signal components above about 45 Hz. In [4], this difficulty was circumvented in learning control applications by applying a zero-phase low-pass filter in the learning control signal. In [5,6] more sophisticated learning control laws are developed that allow one to go beyond this limit without producing bad transient behavior before convergence. Here we wish to adapt the zero-phase filter approach of [4] to the repetitive control problem at hand, as described in the next two sections. An important issue is that the zero phase filter is noncausal, and cannot be applied on-line.

THE BRIDGE BETWEEN REPETITIVE AND LEARNING CONTROL

In learning control the initial conditions are the same for every repetition of the period, i.e., they are reset between repetitions. In pure repetitive control there is no restarting of the system between periods, so that transients can propagate from one period into the next. Also, changes in the control actions during one period can influence the behavior in the next period. It is possible to create control laws that lie between these two situations [12]. Specifically, one could turn on the feedback control system, give it enough time to reach steady state, take data, and compute the repetitive control correction terms (one may choose to make the computations while the system is running, or to restart the system after making the computations). Then start applying this repetitive control signal every period, again waiting until the transients are gone. Then repeat the process form the error data obtained. This procedure eliminates the transient effects discussed above. It does not, however, precisely reduce the repetitive control problem to a learning control problem, because the initial conditions are not repeating. The initial conditions for the repetitions in each case are those developed after transients have died away, producing periodic boundary conditions for the differential equation. Hence, the stability

analysis methods of learning control (e.g., [3]) do not strictly apply. On the other hand, the steady state frequency arguments given above apply directly and justify the approach used here..

ZERO-PHASE LOW-PASS FILTERING IN REPETITIVE CONTROL

A zero-phase filter with a sharp cut-off frequency approximates an ideal filter that has unity gain within the bandwidth of interest and zero elsewhere, and zero-phase shift at all frequencies. This is accomplished by first filtering the signal once using a regular low-pass filter operating forward in time, and then reversing the time history of the data sequence obtained and applying the same filter a second time. The result is a filter of twice the original order and with zero-phase shift. Here, the filter chosen is a fifth order Butterworth filter with a cut-off frequency of 40 Hz, in order to cut off before 90 degree phase lag is reached in the system:

$$F(z) = \frac{a_1 + a_2 z^{-1} + a_3 z^{-2} + a_4 z^{-3} + a_5 z^{-4} + a_6 z^{-5}}{b_1 + b_2 z^{-1} + b_3 z^{-2} + b_4 z^{-3} + b_5 z^{-4} + b_6 z^{-5}} \tag{12}$$

where $a_1 = 2.14 \times 10^{-5}$, $a_2 = 1.07 \times 10^{-4}$, $a_3 = 2.14 \times 10^{-4}$, $a_4 = 2.14 \times 10^{-4}$, $a_5 = 1.07 \times 10^{-4}$, $a_6 = 2.14 \times 10^{-5}$, $b_1 = 1$, $b_2 = -4.19$, $b_3 = 7.07$, $b_4 = -6.01$, $b_5 = 2.57$, $b_6 = -0.44$. The combined effect of the double application is to attenuate signals above the cutoff at 200 dB per decade. Such a filter has initial conditions, and hence it has its own transients. And these initial conditions affect both the beginning and the end of the data length because of the method of applying the filter. To eliminate such effects, three periods of data are used here in the filtering, and then the central period of the resulting filtered signal is extracted. This allows one period for the decay of the filter transients both forward and backward.

The repetitive control law applied here takes the error data, applies the zero-phase low-pass filter to it, and uses the result as the error signal in the repetitive control computation. This process makes the repetitive control blind to error components at frequencies above the 40 Hz cutoff. Hence, the objective of the repetitive control becomes the elimination of as many of the repetitive errors below 40 Hz as possible.

THE EXPERIMENTAL PROCEDURE

This paper presents results of some initial experiments. These reveal some of the important properties of the above approach to applying repetitive control to belt drive systems. In the results reported, learning is attempted only at the 1 second period related to once around for the output shaft, 9 times around for the input shaft, and even harmonics of the idler shaft. Since the desired speed of the output is 1 revolution per second and the sample rate is 1000 Hz, the nominal period for the repetitive controller becomes 1000 samples. The next period is started as soon as the index pulse arrives, so the actual periods may vary several samples. If the output shaft velocity is above nominal, the error data string for that repetition will be short. Updates of the repetitive control signal are make only for the time steps encountered, and if no data has been obtained in previous repetitions for a time step that needs a signal in the current repetition, the desired value is substituted.

A sequence of 30 second runs is made. Each time the shaft starts at zero velocity. For approximately the first three seconds, the shaft is brought up to speed. Then the system starts counting index pulses, and for the first 5 pulses it applies feedback control only. For index pulses 6 to 11 the repetitive control signal computed after the previous 30 second run is applied for each repetition. Data is stored for index pulses 8-11, after roughly two seconds for decay of the transients, and used for FFT analysis in order to understand the improvements made by the repetitive control as a function of frequency. For index pulses 12 -22 the raw p-integrator repetitive controller of equation (2) is applied. The final repetitive control signal obtained is then used for each of the index pulses from 22-24. The resulting three periods of data are fed through the zero-phase low-pass filter, the central period is picked out, and then used to compute the next repetitive control signal. This is applied in the next 30 second run.

The above procedure makes 10 periods of learning from 12 to 22 without using the zero-phase filtering in the updates. Thus, error components above 40 Hz are being amplified. These are then eliminated off line in computing the repetitive signal for the next run. The aim is

to learn faster in this manner. However, the amplification of errors at 90 Hz forces a smaller learning gain (0.1 is used) eliminating some of the anticipated speed up of the convergence.

EXPERIMENTAL RESULTS

The experimental results reported here are limited to addressing two main issues: the ability of the repetitive control to decrease error components for the targeted frequencies, and the influence of the learning process on the frequencies that are not targeted for learning. Artificial sets of sinusoidal disturbances at chosen frequencies are injected to make the study. The repetitive control period is 1 second, and a new period is started each time the index pulse appears. Since the system is restarted for every 30 second run, there is an issue of reproducing the same phase of the error environment with each run. The index pulse is the only information used to determine the starting time of the repetitive control signal. Errors for the output shaft and for the input shaft are then assured to have the same phase. However, the phase of the 4.5 Hz error may change by 180 degrees from run to run. In future experiments, the history of the index pulse from one run to the next will be monitored to know if the current rotation is an even or odd numbered rotation, in order to insure knowing the phase for the 4.5 Hz errors. How to know the phase of the 2.25 Hz harmonics is still under investigation.

In Fig. 3, an output disturbance is injected that consists of a set of sinusoids of amplitude 1×10^{-3}, with frequencies that include all multiples of 4.5 Hz up to 63 Hz. The feedback only graph in Fig. 3 shows the property of the feedback control that it amplifies many frequency components in the command. For feedback control purposes this is acceptable since the command of interest is a constant for this system. After one 30 second run, the errors of the integer valued frequencies have reduced dramatically, for frequencies up to the cutoff. By run 4 only the error component at 36 Hz is still substantial. This is expected due to the large phase lag in the feedback control system at this frequency. These results illustrate the potential of the repetitive control approach to decrease the velocity errors.

The results also show that the 4.5 Hz fundamental and its harmonics are not adversely affected by the repetitive control law period chosen here, which makes the learning process unable to address these frequency components.

Since there are physical reasons to expect other sources of errors that are not related to the output shaft angle, a more detailed study is made to examine how the repetitive control action affects a range of frequencies between those being addressed by the repetitive controller. In Fig. 4, a set of sine wave disturbances are injected with amplitude 1×10^{-3} and frequencies going from 9 Hz to 10 Hz in increments of 0.125 Hz. The amplitudes of these error components are given in the figure for feedback control only, 1st, 2nd and 4th runs. The results indicate substantial improvement in the 9 and 10 Hz error components, and essentially no change in the remaining components.

The sinusoidal disturbances injected above are all simultaneously synchronized at the start with the index pulse, but otherwise operate without regard for the index pulse. In Fig. 5 the sinusoidal disturbance components at multiples of 9 Hz are restarted with each index pulse. This produces a disturbance with period of 1 Hz, but with rich harmonic components. The results show rather fast learning of error components through 27 Hz, with a low residual error level. The error component at 36 Hz learns well initially but fails to approach a small error level. The error component at 45 Hz above the chosen cutoff frequency succeed in some initial improvement, presumably because the low-pass filter cutoff is not perfect. The higher frequency components are essentially unaffected as they should be.

These experiments indicate that a proper implementation of repetitive control could very significantly improve the velocity error in rolling operations, using the simple repetitive control approach of this paper. However, in order to obtain a maximum improvement, it is important that the hardware give sufficient information to know the phase of all periodic error sources that can be addressed by choice of a common period.

CONCLUSIONS

Based on these initial experiments to identify some of the practical issues involved in applying repetitive control to produce precision motion in a double reduction timing belt drive system, we can make the following conclusions:

1.The velocity error of the feedback controller is rich in frequency content, but it is nevertheless possible to find a period to use for the repetitive controller that is also a period for

the vast majority of the velocity error frequency components. Thus, there is a potential for repetitive control to eliminate the majority of the error frequency components.

2. There are error sources that are unrelated to the speed of the output shaft, which will not be periodic with the chosen frequency of the repetitive controller. An example of such an error source is the vibration of the tensioning pulley involving elasticity of the belt. Experiments were conducted to study the effect of the repetitive control action on the error components at such frequencies, and it was determined that the repetitive control action need not have a detrimental effect on these error components.

3. Pure integral control based learning control does not work on the hardware due to the presence of error components at a frequency for which there is a phase lag of 180 degrees through the system.

4. The solution to this difficulty that is used here, is to adapt to the repetitive control problem the zero-phase low-pass filtering concept developed and tested by the authors for learning control applications. This process means that the repetitive controller can address error components for frequencies approximately up to those exhibiting a 90 degree phase lag in the feedback controller, and error components at higher frequencies will not be influenced by the repetitive controller's actions. These components are often rather small due to their high frequency. In the experiments a cutoff frequency of 40 Hz was used.

5. In some situations, it may be important to eliminate error components above this frequency. Future research will develop and test the repetitive control analog of methods for doing this that have been developed by the authors for learning control applications.

6. In order to keep the repetitive control signal synchronized with the error, an index pulse is used on the output shaft. Each time the index pulse arrives a new period is started. This means that the periods vary slightly in length.

7. One would like the repetitive control signal to be one that can be applied immediately, every time the control system is turned on, in order to produce good performance without relearning each time. This is also helpful in the zero-phase filtering since it is non-causal, and must be done in a batch mode. In order for this to work, one must know how to start the repetitive control signal at a time when the phase of all error sources are correct.

8. The use of an index pulse on the output shaft proves not to be sufficient for this purpose. This information alone, is enough to insure knowing the phase for only approximately one fourth of the error peaks in the error FFT. Knowing whether the current index pulse is an odd or even number of pulses since the start of the development of the repetitive control signal can allow one to insure being in phase with approximately one half of the error components. The remaining error components are believed to be associated with bearing frequencies. Research is currently in progress to determine what hardware or data handling is needed in order to know the phase of these error components.

9. The experiments reported here indicate that repetitive control has the potential to significantly decrease the velocity error in belt drive systems. Future experiments will develop methods to know the phase of all error frequency components to be learned, and then determine the actual capability of the method for producing high precision velocity control. Heuristically, the limit is determined by the reproducibility level of the hardware, and the bandwidth used in the low pass filter. Future research will also address the used of methods to increase this bandwidth.

REFERENCES

1. R. H. Middleton, G. C. Goodwin, and R. W. Longman, "A Method for Improving the Dynamic Accuracy of a Robot Performing a Repetitive Task," *International Journal of Robotics Research*, Vol. 8, No. 5, October 1989, pp. 67-74. Also appeared as University of Newcastle, Newcastle, Australia, Department of Electrical Engineering Technical Report EE8546, 1985.

2. M. Tomizuka, T. C. Tsao, K. K. Chew, "Analysis and Synthesis of Discrete-Time Repetitive Controllers," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 111, 1989, pp. 353-358.

3. M. Phan and R. W. Longman, "A Mathematical Theory of Learning Control for Linear Discrete Multivariable Systems," *Proceedings of the AIAA/AAS Astrodynamics Conference*, Minneapolis, Minnesota, August 1988, pp. 740-746

4. H. Elci, M. Phan, R. W. Longman, J.-N. Juang, and R. Ugoletti, "Experiments in the Use of Learning Control for Maximum Precision Robot Trajectory Tracking," Proceedings of the 1994 Conference on Information Sciences and Systems, Princeton, NJ, 1994, to appear.
5. H. Elci, R. W. Longman, M. Phan, J.-N. Juang, and R. Ugoletti, "Discrete Frequency Based Learning Control for Precision Motion Control," Proceedings of the 1994 IEEE International Conference on Systems, Man and Cybernetics, San Antonio, Texas, October 1994, to appear.
6. H. Elci, M. Phan, R. W. Longman, J.-N. Juang, and R. Ugoletti, "Automated Learning Control Through Model Updating for Precision Motion Control," 1994 ASME Winter Annual Meeting, Adaptive Structures and Material Systems Symposium, Nov. 1994, to appear.
7. R. W. Longman and E. J. Solcz, "Small Gain Robustness Issues in the P-Integrator Repetitive Controller," *AIAA/AAS Astrodynamics Conference, A Collection of Technical Papers, Part 2*, Portland, Oregon, August 1990, pp. 537-551.
8. E. J. Solcz and R. W. Longman, "Disturbance Rejection in Repetitive Controllers," *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, Durango, Colorado, August 1991.
9. E. J. Solcz and R. W. Longman, "Robustness of Repetitive Controllers to Frequency Uncertainties," *Proceedings of the 1992 Conference on Information Sciences and Systems*, Department of Electrical Engineering, Princeton University, March 1992.
10. E. J. Solcz and R. W. Longman, "Repetitive Control in the Presence of Multiple Periodic Disturbances," *Proceedings of the 1992 AIAA/AAS Astrodynamics Specialist Conference*, Hilton Head, South Carolina, August 1992.
11. R. W. Longman, and Y.-C. Huang, "Use of Unstable Repetitive Control for Improved Tracking Accuracy," 1994 ASME Winter Annual Meeting, *Adaptive Structures and Material Systems Symposium*, Nov. 1994, to appear.
12. S. C. Lee and R. W. Longman, "Indirect Decentralized Repetitive Control," *Advances in the Astronautical Sciences*, Vol. 82, Part 1, 1993, pp. 93-109.

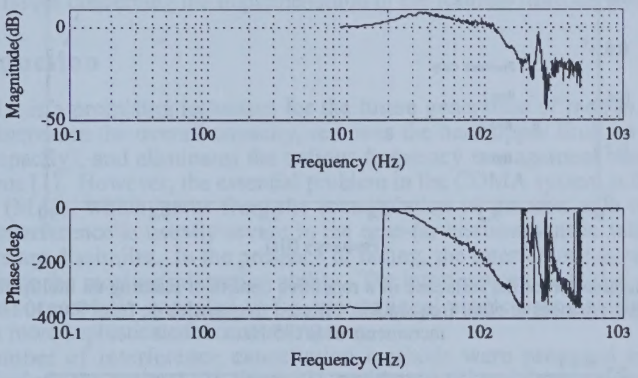


Fig. 1 Magnitude and phase of the closed loop system frequency response.

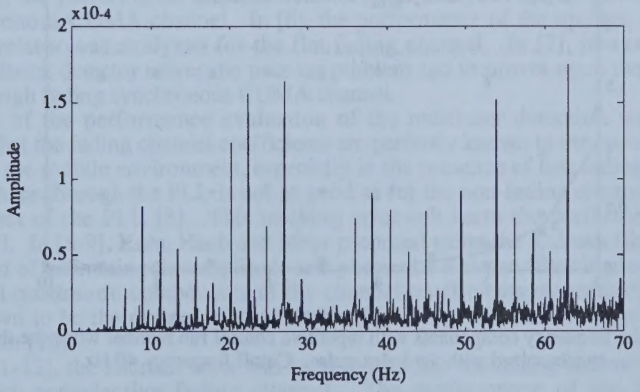


Fig. 2 Frequency spectrum of the velocity error of the feedback control systemthrough 70 Hz.

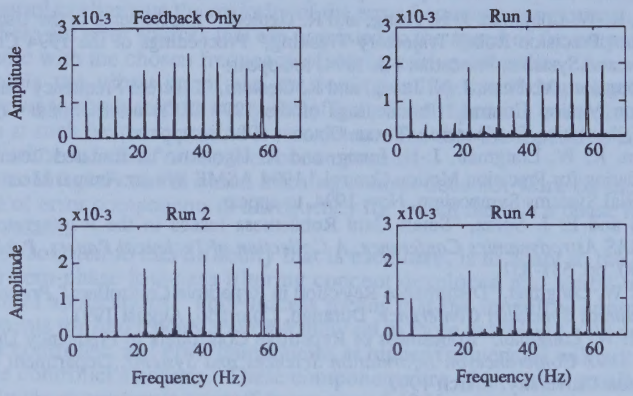


Fig. 3 Improvement in the error spectrum using a repetitive control synchronized to the output shaft index pulse, in eliminating injected sinusoidal disturbances at multiples of 4.5 Hz. Cutoff 40 Hz.

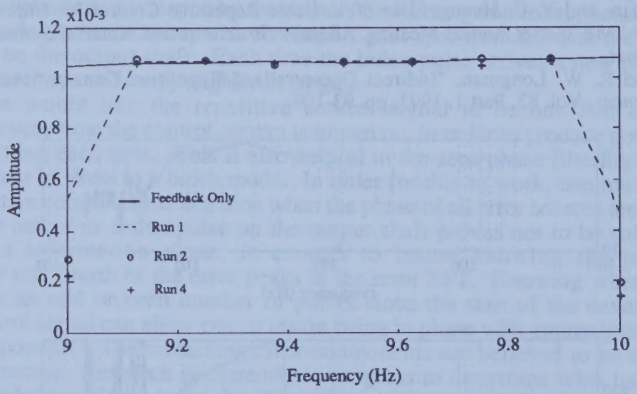


Fig. 4 Detailed examination of the influence of a repetitive controller learning all multiples of 1 Hz, on injected disturbances at other frequencies. Injected frequencies go from 9 to 10 Hz in increments of 0.125 Hz.

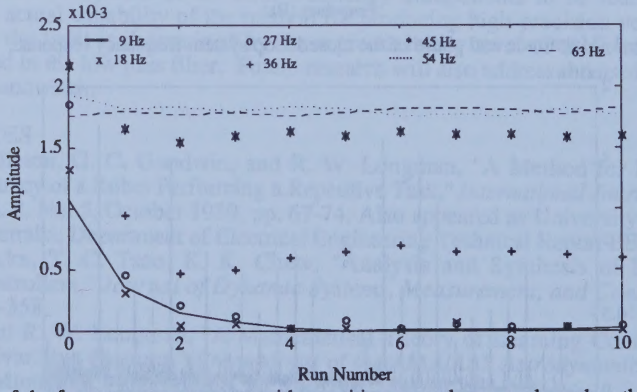


Fig. 5 Decay of the frequency components with repetitive control run number when the disturbance is synchronized with the index pulse. Cutoff frequency 40 Hz.

Channel Estimation and Multiuser Detection for Frequency-Nonselective Fading Synchronous CDMA Channels

Hsin-Yu Wu and Alexandra Duel-Hallen

Department of Electrical and Computer Engineering
North Carolina State University, Raleigh, NC 27695-7911
phone/fax: (919) 515-7352 / 5523
internet: hwu@eos.ncsu.edu, sasha@eos.ncsu.edu

Abstract

Since the Code Division Multiple Access (CDMA) system was proposed for the next generation of mobile communications, several multiuser detection methods have been developed to cope with the Multi-Access Interference inherent in the CDMA system. Fading and multipath characteristics of the mobile radio channel present significant challenges to the designer of multiuser receivers. The fading phenomenon prevents the traditional Phase-Lock Loop (PLL) carrier recovery, so for the coherent detection, more accurate channel estimation is required. In this paper, we consider only frequency non-selective fading channels. Removing the usual assumption that the channel coefficients are perfectly known to the receiver, we model the fading channel as a second order Auto-Regressive (AR) process, and use the decision-directed Kalman filter to estimate the channel coefficients. The effect of estimation errors on the performance of several multiuser detectors is investigated by analysis and simulations. Issues concerning the implementation of the Kalman filter are also addressed.

1. Introduction

CDMA is a promising technique for the future generation of mobile communication systems. It increases the overall capacity, removes the hard upper limit on the numbers of users (soft capacity), and eliminates the tedious frequency management used in the present mobile systems [1]. However, the essential problem in the CDMA system is the Multi-Access Interference (MAI), which stems from the accumulation of the non-zero cross-correlation noise. The interference is usually severe in the near-far environment in which the received signal powers are dissimilar. In the presence of fading, the interference is enhanced and can severely degrade the receiver's detection ability. The interference due to the presence of other users can be successfully canceled if the conventional single-user correlation receiver is replaced by a more sophisticated receiver structure.

A number of interference cancellation methods were proposed recently. These approaches include the optimal [2], linear [3], multistage [4] and decision-feedback detectors [5]. In [3-5], the performance of these detectors was analyzed for additive white Gaussian noise synchronous CDMA channel. In [6], the performance of the optimal detector and the linear decorrelator was analyzed for the flat fading channel. In [7], it was shown that the decision-feedback detector solves the near-far problem and improves upon the decorrelator for the flat Rayleigh fading synchronous CDMA channel.

Most of the performance evaluation of the multiuser detection was based on the assumption that the fading channel coefficients are perfectly known to the base-station receiver. However, in the mobile environment, especially in the presence of fast fading, the tracking of the carrier phase through the PLL is not as good as for the non-fading channel because of the hang-up effect of the PLL [8]. This tracking error can harm the performance of the PSK systems [8-9]. In [8-9], Kam, Haeb and Meyr proposed using the Kalman filter to replace the PLL. Instead of estimating the amplitude and phase, the Kalman filter is used to estimate the In-phase and quadrature components of the channel coefficients respectively. The Kalman filter is known to be the optimum MMSE estimator if the channel coefficients obey Gauss-Markov model [10].

In [11-12], the Kalman filter was used to estimate the fading channel coefficients for the frequency-nonselective fading channel. The performance of the linear multiuser decorrelator was evaluated assuming that the correct decisions are available for the decision-

directed estimator. The BER was shown to be heavily dependent on the estimation error. In this paper, we also concentrate on the frequency-nonselective fading channel. We examine the effect of the estimation error and the actual decisions on the performance of the decorrelator, the two-stage detector and the decision-feedback detector.

The paper is arranged as follows. In section 2, we review the synchronous CDMA model and several multiuser detector structures. In section 3, we formulate the signal model of the flat fading channel, and show how the Kalman filter can be used to estimate the channel coefficients. Some implementation issues are discussed. The theoretical BER for multiuser detectors and the actual simulation results are presented in section 4.

2. Synchronous CDMA System Model and Multiuser Detectors

Consider a synchronous CDMA system with K users for a Rayleigh fading channel. The received signal is given by

$$r(t) = \sum_{i=0}^{\infty} \sum_{n=1}^{K} C_n(i)\, b_n(i)\, s_n(t - iT) + n(t),$$

where $C_n(i)$ is the complex fading coefficient, $b_n(i)$ is the information bit of the n-th user drawn from $\{+1,-1\}$, $s_n(t)$ is the normalized signature waveform of the n-th user (the energy of $s_n(t)$ is one), and n(t) is the additive white Gaussian noise with power N_0.

The front-end of a multiuser detector consists of a bank of matched filters. Each matched filter is designed to correlate each individual user's signature waveform. The output of the matched filter bank at the k-th sampling instance can be represented by the column vector **y** of length K:

$$\mathbf{y}(k) = \mathbf{R}\mathbf{W}(k)\mathbf{b}(k) + \mathbf{n}(k), \qquad (1)$$

where the input vector **b**(k) consists of the information bits of K users. The matrix **R** has components $\mathbf{R}_{l,m} = \int_{-\infty}^{\infty} s_l(t)\, s_m(t)\, dt$ (l,m = 1, ... ,K), the cross-correlations between the signature waveforms of users l and m. The diagonal matrix **W**(k) has components $\mathbf{W}_{n,n} = \sqrt{E_{b_n}}\,|C_n(k)|\exp(-j\theta_n(k))$ (n=1,...,K), where Eb_n is the bit energy of user n. The vector **n**(k) is a K-dimensional complex Gaussian noise vector. We assume that the channel coefficients $|C_n(k)|\exp(-j\theta_n(k))$ are constant during the transmission interval, and are generated by independent complex Gaussian random processes. The amplitude $|C_n(k)|$ is Rayleigh distributed, and the phase $\theta_n(k)$ is uniform on $[0,2\pi)$.

We consider three multiuser detectors: the decorrelator, the two-stage detector and the decision-feedback detector. The linear decorrelator [3] can be obtained by multiplying the signal (1) by the inverse of the correlation matrix **R**:

$$\mathbf{z}(k) = \mathbf{R}^{-1}\mathbf{y}(k) = \mathbf{W}(k)\mathbf{b}(k) + \mathbf{v}(k), \qquad (2)$$

where the noise vector $\mathbf{v}(k) = \mathbf{R}^{-1}\mathbf{n}(k)$. The decision for the n-th user is made by $\hat{b}_n(k) = sgn(\mathrm{Re}[z_n(k)\,\hat{C}_n^*(k)])$, where $\hat{C}_n(k)$ is the estimate of the channel coefficient $C_n(k)$. The output of the decorrelator for the n-th user, $z_n(k)$, contains no interference from other users; however, the colored Gaussian noise **v**(k) is enhanced.

The two-stage detector considered here uses the decorrelator as the first stage [4]. The tentative decisions obtained from the output of the decorrelator are then used to reconstruct the MAI in the second stage. The reconstructed MAI is subtracted from the output of the matched filter bank. The final decisions are made after canceling the interference.

The decision-feedback detector first arranges the users in the descending energy order: $|\mathbf{W}_{1,1}(k)| \geq |\mathbf{W}_{2,2}(k)| \geq \ldots \geq |\mathbf{W}_{K,K}(k)|$ (the estimates of the energies are used in the actual implementation), and then performs the Cholesky factorization of the matrix **R**(k) to find the noise-whitening filter $(\mathbf{F}(k)^T)^{-1}$ (i.e., $\mathbf{R}(k)=\mathbf{F}^T(k)\mathbf{F}(k)$) [5,7]. Note that the matrix **R**(k) varies with the sampling time k depending on the energy order. This was not the case for the

decorrelator and the two-stage detector. If the noise-whitening filter is applied to the matched filter output y(k), the resulting output vector is

$$\tilde{\mathbf{y}}(k) = \mathbf{F}(k)\mathbf{W}(k)\mathbf{b}(k) + \tilde{\mathbf{n}}(k), \tag{3}$$

where $\mathbf{F}(k)$ is a left lower triangular matrix with its entries denoted by $f_{ij}(k)$, and $\tilde{\mathbf{n}}(k)$ is white complex Gaussian noise vector. The lower triangular structure of $\mathbf{F}(k)$ leads to the successive interference cancellation. The signal of each user in (3) is free from interference due to weaker users. The first user (the strongest user) can make the decision first since it does not contain any interference from the other users ($\tilde{y}_1(k) = f_{11}(k)\sqrt{E_{b_1}}\, C_1(k) b_1(k) + \tilde{n}_1(k)$). Using the decision of the first user, the second strongest user can cancel the interference due to the first user. Similarly, the multiuser interference in the signal of the n-th user can be completely deleted if decisions of the users 1 to n-1 are correct.

3. Channel Estimation

Channel estimation is an essential problem for the coherent detection. For the three detectors described in section 2, the estimates $\hat{C}_n(k)$ are required to make reliable decisions. In particular, for mobile communications where channel varies rapidly, the performance of a receiver depends heavily upon the accuracy of the channel estimation. We use the Kalman filter, which, unlike the LMS and the RLS algorithms, incorporates the channel dynamics to compute the least mean square estimation error and the channel estimates.

Signal model of a fading channel and the Kalman filter

The power spectrum of a Rayleigh flat fading channel contains narrow band spectral peaks upon the maximum Doppler frequencies [13]. A simple signal model which describes the spectral peak behavior is a lightly damped second order AR process [14]. Once the fading channel model is determined, the Kalman filter is ready to implement.

For user n (n is not shown for simplicity), the discrete-time signal model of the flat fading channel coefficient is given by a second order Auto-Regressive (AR) process:

$$C(k) = -a_1 C(k-1) - a_2 C(k-2) + w(k), \tag{4}$$

The 2nd order AR process can also be expressed as the Gauss-Markov model,

$$\mathbf{X}(k+1) = \mathbf{F}\,\mathbf{X}(k) + \mathbf{G}\, w(k), \tag{5}$$

where $\mathbf{X}(k) = [C(k)\ C(k-1)]^T$ is the state variable, $\mathbf{F} = \begin{bmatrix} -a_1 & -a_2 \\ 1 & 0 \end{bmatrix}$ and $\mathbf{G} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ are model parameter matrices. The measurement used to estimate the channel coefficients is the output of the decorrelator for user n. This signal can be expressed as

$$z(k) = \sqrt{E_b}\, b(k) C(k) + v(k) = \mathbf{H}'(k)\,\mathbf{X}(k) + v(k), \tag{6}$$

where $\mathbf{H}(k) = \sqrt{E_b}\, b(k) [1 \quad 0]^T$, and v(k) is the Gaussian noise at the output of the decorrelator with variance $N_0[\mathbf{R}^{-1}]_{n,n}$.

Once the fading model (5) and the measurements (6) are acquired, the Kalman filter can be used as the channel estimator. The estimates are generated recursively by the following equations [10]:

$$\mathbf{M}(k) = \Sigma(k)\,\mathbf{H}(k)\,[\mathbf{H}'(k)\,\Sigma(k)\,\mathbf{H}(k) + R]^{-1}, \tag{7}$$

Kalman filter gain:
$$\mathbf{K}(k) = \mathbf{F}\,\mathbf{M}(k), \tag{8}$$

Prediction:
$$\hat{\mathbf{X}}(k+1) = [\mathbf{F} - \mathbf{K}(k)\,\mathbf{H}'(k)]\,\hat{\mathbf{X}}(k) + \mathbf{K}(k)\, z(k), \tag{9}$$

Error Covariance:
$$\Sigma(k+1) = \mathbf{F}\,[\Sigma(k) - \mathbf{M}(k)\,\mathbf{H}'(k)\,\Sigma(k)]\,\mathbf{F}' + \mathbf{G}\,Q\,\mathbf{G}' \text{ (Ricatti equation)}, \tag{10}$$

where R and Q are the variances of the excitation noise and channel noise, i.e., $R = E\left(|v(k)|^2\right)$, $Q = E\left(|w(k)|^2\right)$.

The structure of the decorrelator with the Kalman channel estimator is shown in Figure 1. It is convenient to use the output of the decorrelator z(k) as the input of the Kalman filter

since it does not contain MAI shown in (2). The measurement z(k) is the same as for the single user case except for the enhanced noise. Therefore, the channel estimate of each user can be obtained separately, i.e., joint estimation is not necessary. In [12], the outputs of the matched filter bank were used to estimate the channel coefficients; however, in this case, the joint estimation is required. The joint estimation involves the irreducible data dependency in updating the Kalman filter gain (8), the channel estimate (9), and the error covariance (10) [12]. The matched filter bank is prone to make decision errors when the interference is high, so the unreliable decisions will lead to the inaccurate estimation. Compared with the non-joint estimation, it was shown in [15] that the joint estimation has only limited improvement in the estimation error in the high SNR region.

Implementation Issues

Equations (7-10) show that the channel estimates depend on the matrices **F**, **G** and **H**(k). The matrix **H**(k) depends on the unknown data bit b(k). When the estimator is in the training mode, the correct data bits are available. In the decision-directed mode, the decision $\hat{b}(k)$ is used in place of b(k). In this section, we will discuss the details of the Kalman filter implementation.

(i) The Training Mode:

The main purpose of the training mode is to let the initial channel estimates converge to the channel coefficients (9). The error covariance matrix can also be computed at this time, and can later be used in the decision-directed mode. From (10), the error covariance matrix $\Sigma(k)$ appears to be data-dependent since the matrix $\Sigma(k)$ is dependent on the matrix **H**(k). However, by examining (7) and (10) carefully, we observe that in our case, the product **M**(k)**H**(k) is given by the quadratic form which eliminates the data-dependency. Using this fact, we can show by a simple induction argument that the error covariance matrix $\Sigma(K)$ does not depend of the transmitted data stream. This also implies that in the decision-directed mode, wrong decisions will not affect the error covariance matrix. Therefore, in principle, the steady state error covariance matrix can be computed off-line prior to the reception, and used in the decision-directed mode to save the computational load.

(ii) Decision-directed mode:

In the decision-directed mode, the data-dependent matrix **H**(k) is unknown, so it is replaced by $\hat{\mathbf{H}}(k) = \sqrt{E_b}\,\hat{b}(k)\,[1 \quad 0]^T$, where $\hat{b}(k)$ is the decision. The substitution modifies (7-9). The prediction equations are now given by (11-13),

$$\hat{\mathbf{M}}(k) = \Sigma(k)\,\hat{\mathbf{H}}(k)\,[\hat{\mathbf{H}}'(k)\,\Sigma(k)\,\hat{\mathbf{H}}(k) + R]^{-1} \tag{11}$$

Kalman filter gain:
$$\hat{\mathbf{K}}(k) = \mathbf{F}\,\hat{\mathbf{M}}(k) \tag{12}$$

Prediction:
$$\hat{\mathbf{X}}(k+1) = [\mathbf{F} - \hat{\mathbf{K}}(k)\,\hat{\mathbf{H}}'(k)]\,\hat{\mathbf{X}}(k) + \hat{\mathbf{K}}(k)\,z(k) \tag{13}$$

Note that this does not affect the steady-state covariance matrix as discussed in the previous section. One should note that the stationarity of the error covariance does not imply that the "measured" mean square error (MSE) ($\frac{1}{N}\sum_{k=1}^{N}\left|C(k) - \hat{C}(k)\right|^2$) approaches the theoretical MSE $\Sigma(k)_{1,1}$. Actually the measured MSE sometimes may be much larger than the theoretical MSE. When the measured MSE is much larger than the theoretical one, an interesting "reversal" phenomenon occurs. The term "reversal phenomenon" refers to the fact that the estimates are close to the opposites of the actual channel coefficients (phase shift of π). It is illustrated in Figure 2. The channel estimates and channel coefficients are reversed between iteration 300 and 600. During this interval, almost all the decisions are incorrect. Observations show that the reversal phenomenon starts during the deep fades, i.e., when the magnitudes of the channel coefficients are quite small. In the deep fades, the decision errors are likely to occur, and the Kalman estimator will produce the inaccurate estimates, which in turn will cause the subsequent decision errors.

The following arguments explain the reversal phenomenon. Substituting the output of the decorrelator (6) into (13) (ignoring the noise term), we have

$$\hat{\mathbf{X}}(k+1) = \mathbf{F}\hat{\mathbf{X}}(k) + \hat{\mathbf{K}}(k)\,\{\,\mathbf{H}(k)\,\mathbf{X}(k) - \hat{\mathbf{H}}(k)\,\hat{\mathbf{X}}(k)\,\}. \tag{14}$$

Suppose the decision is wrong, i.e., $\hat{b}(k) = -b(k)$. Then $\hat{\mathbf{H}}(k) = -\mathbf{H}(k)$, and (14) becomes

$$\hat{\mathbf{X}}(k+1) = \mathbf{F}\,\hat{\mathbf{X}}(k) - \hat{\mathbf{K}}(k)\,\hat{\mathbf{H}}(k)\{\,\mathbf{X}(k) + \hat{\mathbf{X}}(k)\,\}. \tag{15}$$

The sum of the actual channel coefficient state vector X(k) (5) (without the noise term w(k)) and the estimate (15) gives

$$\mathbf{X}(k+1) + \hat{\mathbf{X}}(k+1) = \{\,\mathbf{F} - \hat{\mathbf{K}}(k)\,\hat{\mathbf{H}}(k)\,\}\{\,\mathbf{X}(k) + \hat{\mathbf{X}}(k)\,\} \tag{16}$$

If the Kalman filter is stable, i.e., the magnitudes of all the eigenvalues of the matrix $\mathbf{F} - \hat{\mathbf{K}}(k)\,\hat{\mathbf{H}}(k)$ are less than one, $\mathbf{X}(k) \to -\hat{\mathbf{X}}(k)$ as k increases. Thus, the sequence of the incorrect decisions may trigger the reversal phenomenon.

On the other hand, if the decision is correct, then $\hat{\mathbf{H}}(k) = \mathbf{H}(k)$, (14) becomes

$$\hat{\mathbf{X}}(k+1) = \mathbf{F}\,\hat{\mathbf{X}}(k) + \hat{\mathbf{K}}(k)\,\hat{\mathbf{H}}(k)\{\,\mathbf{X}(k) - \hat{\mathbf{X}}(k)\,\} \tag{17}$$

Subtracting (17) from (5), we obtain

$$\mathbf{X}(k+1) - \hat{\mathbf{X}}(k+1) = \{\,\mathbf{F} - \hat{\mathbf{K}}(k)\,\hat{\mathbf{H}}(k)\,\}\{\,\mathbf{X}(k) - \hat{\mathbf{X}}(k)\,\} \tag{18}$$

Equation (18) implies that a sequence of correct decisions will result in the estimate $\hat{\mathbf{X}}(k)$ approaching the actual channel coefficient X(k) provided that the ignored noise terms are small. These arguments and the experimental results show that a few consecutive errors in the deep fades are likely to trigger the reversal phenomenon. Once the reversal of the channel coefficients is established, the incorrect decisions continue to occur, even when the channel is "outside" the deep fades. This is due to the fact that the decision is given by $\hat{b}(k) = sgn(\mathrm{Re}[z(k)\,\hat{C}^{*}(k)])$. When the estimate $\hat{C}(k) \approx -C(k)$, the decision is likely to be $\hat{b}(k) = -b(k)$. The reversal typically continues until another deep fade during which several "decision errors" (which are actually the correct decisions) may disrupt the convergence in (16) and allow the estimator to recover back to (18).

To avoid the reversal, we set up a threshold level P. When the amplitude of the current channel estimate becomes smaller than the threshold level ($|\hat{C}(k)| < P$), i.e., the channel is in a deep fade, the transmitter is informed to send a training bit next time. The transmission throughput is related to the threshold level. The higher the threshold level, the lower the throughput.

4. Performance Analysis

The fading channel model is given by a second order AR process with a1=-1.9935, a2=0.996. The model corresponds to a second order IIR filter with poles given by $r_d e^{\pm j\omega_d T}$, where rd=0.998 and normalized fading rate $\omega_d T = 2\pi f_d T = 0.0503$ [14]. The data rate is 10 kb/s and maximum Doppler shift (f_d) is 80 Hz.

The synchronous CDMA system has 2 users. We consider two cases. In the first case, the MAI is moderate, and the cross-correlation between the signature waveforms r=0.3. In the second case, the MAI is strong, r=0.9. We examine the performance of the multiuser detectors with Kalman channel estimators by computing the theoretical limits and by simulations.

Theoretical Results

When the channel coefficients are estimated using the MMSE criterion, the BER of the decorrelator for user n was given by [11]:

$$PE_{dec,BPSK} = \frac{1}{2}\left[1 - \sqrt{\frac{1-\Gamma_n}{1+[\mathbf{R}^{-1}]_{n,n}/\bar{\gamma}_n}}\right], \tag{19}$$

where $[\mathbf{R}^{-1}]_{n,n}$ is the n-th diagonal element of the inverse of the cross correlation matrix, $\bar{\gamma}_n$ is the average SNR defined by $Eb_n\ E(|C_n(k)|^2)/N_0$, and Γ_n is the normalized estimation error variance for the n-th user, defined by

$$\Gamma_n = \frac{E\left(\left|C_n(k)-\hat{C}_n(k)\right|^2\right)}{E\left(\left|C_n(k)\right|^2\right)}$$

The steady-state estimation error variance can be calculated by the Ricatti equation (10). From the orthogonality principle of the MMSE estimator, i.e., $E\left(\hat{C}(k)e^*(k)\right)=0$ where $e(k)=C(k)-\hat{C}(k)$, we obtain

$$E\left(C(k)\hat{C}^*(k)\right)=E\left(\hat{C}(k)\hat{C}^*(k)\right)=(1-\Gamma)E\left(|C(k)|^2\right) \tag{20}$$

Equation (20) implies that the normalized covariance matrix of [C(k) $\hat{C}(k)$] is $\begin{bmatrix} 1 & 1-\Gamma \\ 1-\Gamma & 1-\Gamma \end{bmatrix}$.

Assuming that C(k) and $\hat{C}(k)$ are jointly Gaussian, we obtain the vector

$$\begin{bmatrix} C(k) \\ \hat{C}(k) \end{bmatrix} = \mathbf{A}\begin{bmatrix} w_1(k) \\ w_2(k) \end{bmatrix} \tag{21}$$

where $w_1(k)$ and $w_2(k)$ are i.i.d. while Gaussian random variables, and $\mathbf{AA}^T=\begin{bmatrix} 1 & 1-\Gamma \\ 1-\Gamma & 1-\Gamma \end{bmatrix}$.

Using (21) and the Monte Carlo simulation technique, we generate the channel coefficients C(k) and the corresponding channel estimates $\hat{C}(k)$, and calculate the BER of the decorrelator, the 2-stage and the decision-feedback detectors. The simulation results for the decorrelator agree with (19).

To compare with the differentially coherent modulation scheme for the fading channel, the decorrelator with DPSK is considered [12, 15]. The BER of this decorrelator is [12]

$$PE_{dec,DPSK} = \frac{1}{2}\left[1-\frac{\rho}{1+[\mathbf{R}^{-1}]_{n,n}/\bar{\gamma}_n}\right] \quad \text{where} \quad \rho=\frac{E\left(C(k)C^*(k-1)\right)}{E\left(|C(k)|^2\right)}.$$

Figures 3 and 4 show the theoretical BER of these multiuser detectors with Kalman channel estimators for the two user channels with r=0.3 and r=0.9 receptively. With moderate MAI (r=0.3), the decorrelator, the 2-stage and the decision-feedback detector have similar performance which is close to the single user system with the same imperfect channel estimation. For high MAI (r=0.9), the 2-stage and the decision-feedback detectors outperform the decorrelator even when compared with the decorrelator with perfect channel estimation for SNR below 25dB. In high SNR region, the BER of all three detectors are similar. This probability error floor is mainly due to the irreducible mean square estimation error (Γ_n in (19)) rather than to the MAI [12, 15]. We also observe that multiuser detectors outperform the decorrelator with DPSK in both cases. Note that the BER of the three multiuser detectors are higher than the BER of the single user system. This is due to the fact that the noise component in the input to the channel estimator for the single user system is not enhanced as for the multiuser detectors. Therefore, the single user system has smaller estimation error and lower BER. In summary, the performance gains of the 2-stage and the decision-feedback detectors are still preserved. However, the improvement with respect to the decorrelator is not as significant as for the perfect estimation case.

Actual Simulation Results

The error covariance matrix used in the decision-directed mode is computed during training (about 100 iterations) and could be computed off-line. The threshold level is set in the following way. Suppose that the amplitude of the channel coefficient and the channel estimate are approximately the same. Using (6), we can compute the probability of error when the amplitude of the channel coefficient |C|=P. We choose the threshold level P so that this

probability is 10^{-2}. Thus, we avoid the situations when the channel undergoes deep fades. For example, for the two-user case with cross-correlation r, this criterion leads to $Q(\sqrt{\frac{P^2 E_b}{N_0/2}(1-r^2)}) = 10^{-2}$. If the amplitude of the current estimate $|\hat{C}(k)|$ <P, the next bit b(k+1) sent by the transmitter is used for retraining. Although this is not a realistic ARQ protocol, it gives an upper bound on the throughput (the ratio of the number of data bits to the number of all transmitted bits (data+training)). Table 1 shows the throughputs given this threshold criterion. Note that the throughput of the decorrelator with r=0.9 is much lower than the throughput of the single user system in the low SNR region. This is due to the noise enhancement in the output of the decorrelator.

Average SNR (dB)		10	15	20	25	30	35
Throughput(%)	Single user	86.9	95.8	98.6	99.5	99.8	99.9
	2 user (r=0.3)	86.1	95.3	98.5	99.5	99.8	99.9
	2 user (r=0.9)	47.3	79.6	92.9	97.7	99.3	99.7

Table 1: Throughputs of the single user system and the decorrelator (2 user) for the simulation shown in Figure 5.

Figure 5 shows the actual simulation results. Compared with the theoretical results, the BER are lower because the receivers do not make decisions during deep fades. The BER of the decorrelator (r=0.3) is very close to that of the single user system. We did not encounter any reversals during the simulation with 10^6 bits. However, the probability of reversal is still under investigation.

5. Work in Progress

In this paper, we use the Kalman filter as the channel estimator for the Rayleigh flat fading channel. The error propagation problem (termed the "reversal phenomenon") present in the decision-directed mode was found to be a major obstacle to the reliable performance of this estimator. To prevent this problem, we set up a threshold to avoid the deep fades. Another method, the DPSK with coherent detection, is under the investigation. This is due to the fact that the reversal (phase shift of π) will not affect the difference between the phases of two consecutive bits. Our future study will concentrate on practical considerations, such as applying Kalman filter and other estimation methods to the Jakes' fading model, and investigating the performance of multiuser detectors for asynchronous multipath fading channels.

Acknowledgments and Support

The authors would like to acknowledge Prof. Tom Alexander for valuable discussions. Support for this work was provided by the ECE Department and the Center for Communications and Signal Processing of the North Carolina State University.

References

[1] W.C.Y. Lee, "Overview of Cellular CDMA," *IEEE Transactions on Vehicular Technology*, VOL. 40, N0. 2, May 1991.

[2] S. Verdu, "Minimum Probability of Error for Asynchronous Gaussian Multiple-Access Channel," *IEEE Transactions on Information Theory*, Vol. IT-32, No. 1, Jan. 1986, pp. 85-96.

[3] R. Lupas, S. Verdu, "Linear Multiuser Detectors for Synchronous Code-Division Multiple-Access Channel," *IEEE Transactions on Information Theory*, Vol. IT-35, No. 1, Jan. 1989, pp. 123-136.

[4] M. K. Varanasi, B. Aazhang, "Near-optimum Detection in Synchronous Code-Division Multiple Access Channel," *IEEE Transactions on Communications*, Vol. COM-39, No. 5, May 1991, pp. 725-736.

[5] A. Duel-Hallen, "Decorrelating Decision-Feedback Multiuser Detector for Synchronous Code-Division Multiple Access Channel," *IEEE Transactions on Communications*, Vol. COM-41, No. 2, Feb. 1993, pp.285-290.

[6] Z. Zvonar, D. Brady, "Multiuser Detection in Single-Path Fading Channels," submitted to *IEEE Transactions on Communications*.

[7] H-Y Wu, A. Duel-Hallen, "Performance of Multiuser Decision-Feedback Detectors for Flat Fading Synchronous CDMA Channels," *Proc. of the 28th Annual Conference on Information Sciences and Systems*, Princeton University, Princeton, NJ, March 1994.

[8] R. Haeb, H. Meyr, "A systematic Approach to Carrier Recovery and Detection of Digitally Phase Modulated Signals on Fading Channels," *IEEE Transactions on Communications*, VOL. 37, NO. 7, July 1989, pp748-754.

[9] P.Y. Kam, "Optimal Detection of Digital Data Over the Nonselective Rayleigh Fading Channel with Diversity Reception," *IEEE Transactions on Communications*, VOL. 39, NO. 2, February 1991, pp214-219.

[10] B.D.O.Anderson and J.R.Moore, *Optimal Filtering*, NJ: Prentice-Hall, 1979.

[11] Z. Zvonar, D. Brady, "A Comparison of Differentially Coherent and Coherent Multiuser Detection With Imperfect Phase Estimates in a Rayleigh Fading Channel," *Proceedings of the 1993 International Symposium on Information Theory*, p.48

[12] Z. Zvonar, M. Stojanovic, "Performance of Multiuser Diversity Reception in Nonselective Rayleigh Fading CDMA Channels," submitted to CTMC '94 (Globecom'94).

[13] W.C.Jakes, Jr., *Microwave Mobile Communication*, John Wiley &Sons, New York, 1974, p75.

[14] L. Lindbom, "Simplified Kalman Estimation of Fading Mobile Radio Channels: High Performance at LMS Computational Load," *IEEE ICASSP*, VOL. 3, pp352-355 April, 1993.

[15] Z. Zvonar, *Multiuser Detection for Rayleigh Fading Channels*, Ph.D. Thesis, Chapter 4, Northeastern University, Boston, Massachusetts, 1993.

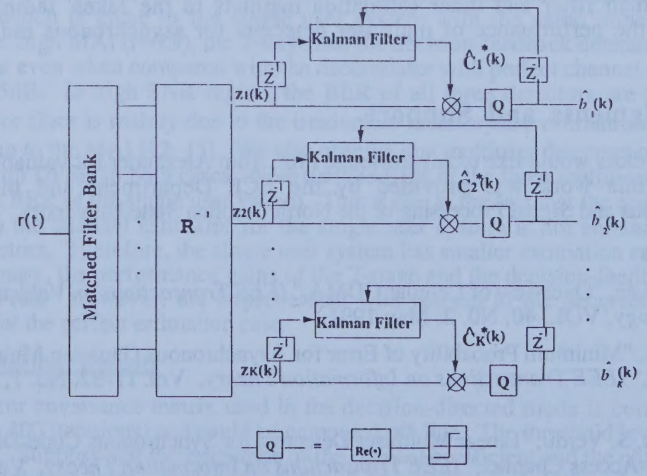


Figure 1: The decorrelator with the Kalman estimators

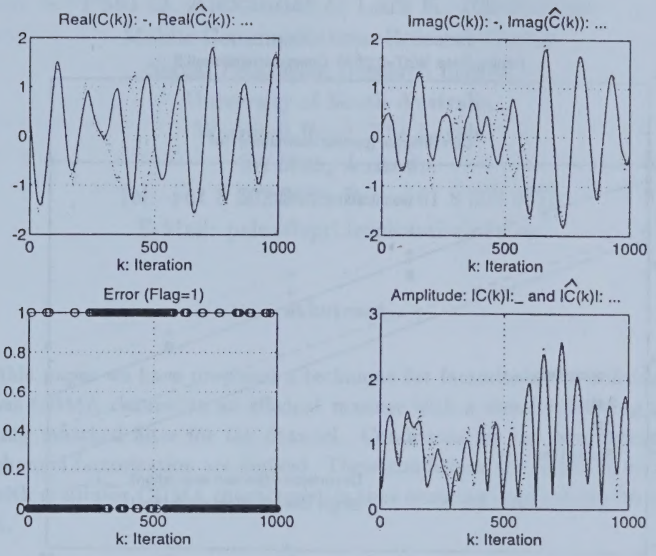


Figure 2: The reversal phenomenon of the Kalman estimator in the decision-directed mode

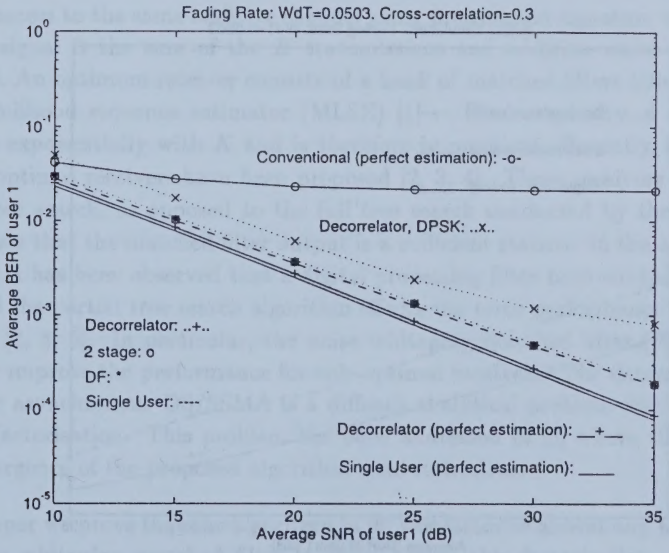


Figure 3: BER of multiuser detectors for a two-user channel (r=0.3)

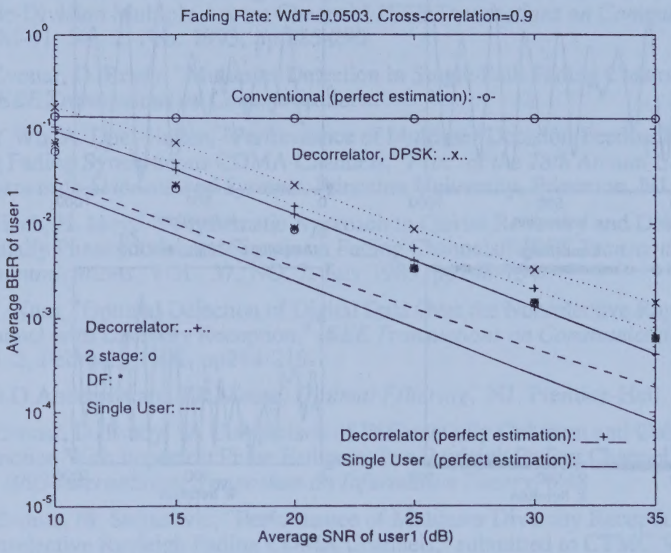


Figure 4: BER of multiuser detectors for a two-user channel (r=0.9)

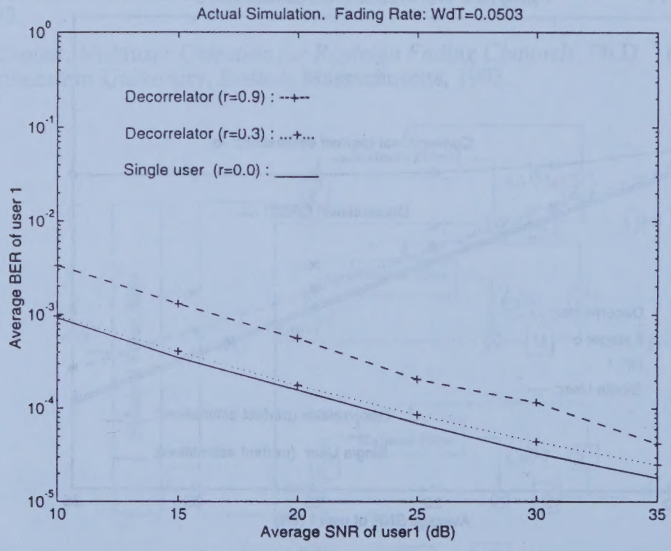

Actual Simulation. Fading Rate: WdT=0.0503
Decorrelator (r=0.9) : --+--
Decorrelator (r=0.3) : ...+...
Single user (r=0.0) : _____
Average BER of user 1
Average SNR of user1 (dB)

Figure 5: Actual BER of the decorrelator with Kalman estimator using the threshold criterion

Factorising the Asynchronous CDMA Channel

Paul D. Alexander & Lars K. Rasmussen
Mobile Communications Research Centre
Signal Processing Research Institute
University of South Australia
Warrendi Road, The Levels
SA 5095, Australia
Tel: +61 8 302 3885, Fax: +61 8 302 3873
E-Mail: palex@spri.levels.unisa.edu.au

Abstract

In this paper we have proposed a technique for factorising the multiuser asynchronous CDMA channel in an efficient manner with a view to deriving the noise whitening matched filter for the channel. Conditions for the convergence to the exact channel factorisation are derived. These conditions are shown to coincide exactly with multiuser CDMA characteristics thus ensuring applicability to multiuser CDMA.

1 Introduction

In a multiuser asynchronous code–division multiple–access (CDMA) system K users have random access to the same additive channel using preassigned signature waveforms. The received signal is the sum of the K transmissions and additive white Gaussian noise (AWGN). An optimum receiver consists of a bank of matched filters followed by a maximum likelihood sequence estimator (MLSE) [1] . The complexity of such a receiver increases exponentially with K and is therefore impractical. Recently, linear complexity, sub–optimal receivers have been proposed [2, 3, 4]. These receivers are based on a partial tree search, as opposed to the full tree search conducted by the MLSE. It has been shown that the matched filter output is a sufficient statistic in the optimal case [1]. However, it has been observed that a digital processing filter between the matched filter bank and the partial tree search algorithm effects the error performance in sub–optimal receivers [2, 3, 5]. In particular, the noise whitening matched filter (WMF) has been shown to improve the performance for sub–optimal receivers. The determination of the WMF for asynchronous DS/SSMA is a difficult analytical problem due to the required channel factorisation. This problem has been addressed in [6] where the condition for the convergence of the proposed algorithm were restrictive.

In this paper we prove that the algorithm in [6] will factorise almost any CDMA channel. The noise whitening matched filter is derived from this factorisation. For the asynchronous case we show that the WMF is determined from the Cholesky decomposition

[7] of $\boldsymbol{Y}$, where $\boldsymbol{Y}$ is the solution to the matrix equation $\boldsymbol{Y} = \boldsymbol{R}_0 - \boldsymbol{R}_1^{\mathrm{T}}\boldsymbol{Y}^{-1}\boldsymbol{R}_1$. A recursive algorithm is proposed to determine $\boldsymbol{Y}$. Due to its recursive nature, the algorithm it is adaptive in terms of changes in the correlations of the users, and in the number of users. The convergence properties of the algorithm are studied and it is shown that the algorithm will convergence for any non–singular CDMA channel.

The paper is organised as follows. In Section 2 the discrete time channel model for DS/SSMA is described and the factorisation problem is defined in Section 3. The algorithm used to solve the factorisation problem is presented in Section 4. The convergence properties of the algorithm are studied in Section 5.

2 System Model

An asynchronous DS/SSMA system with K users in an AWGN channel is considered. Each user has a pre–assigned signature waveform, $s_k(t)$, defined by their spreading code. The power of these signature waveforms is $\int_0^T s_k^2(t)dt = 1$. The transmitted data for user k at symbol interval i is $b_k(i)$, chosen uniformly from the set $\{-1, +1\}$. BPSK is used as the modulation technique, and the transmitted baseband signal for user k during some symbol interval i is therefore $b_k(i)s_k(t - iT)$. We assume these K signals arrive at the receiver with independent delays $\tau_k \in [0, T)$, and posses energies $w_k \leq 1$. The users are numbered, without loss of generality such that $0 \leq \tau_1 \leq \tau_2 \leq \cdots \leq \tau_K < T$. The AWGN is described by the process $n(t)$ having power spectral density N_0. The received signal in the baseband is therefore

$$r(t) = \sum_{i=0}^{M}\sum_{k=1}^{K} b_k(i)\sqrt{w_k}\; s_k(t - iT - \tau_K) + n(t),$$

where M symbols are transmitted. The received signal is passed through a bank of K matched filters. Each filter is matched to a particular users signature waveform and the output of the filter is sampled at the end of the particular user's symbol period as shown in Fig. 1.

At symbol interval i the output of the filter matched for user k is

$$y_k(i) = \int_{-\infty}^{+\infty} r(t)\; s_k(t - iT - \tau_k)dt.$$

The modulating system, the physical channel, and the matched filters can be modelled as a discrete time channel (DTC). The DTC input is represented by a K element vector using the unit delay operator D as

$$\boldsymbol{b}(D) = \left\{\sum_{i=0}^{M} b_k(i)D^i\right\}_1^K.$$

The DTC output is similarly represented as

$$\boldsymbol{y}(D) = \left\{\sum_{i=0}^{M} y_k(i)D^i\right\}_1^K.$$

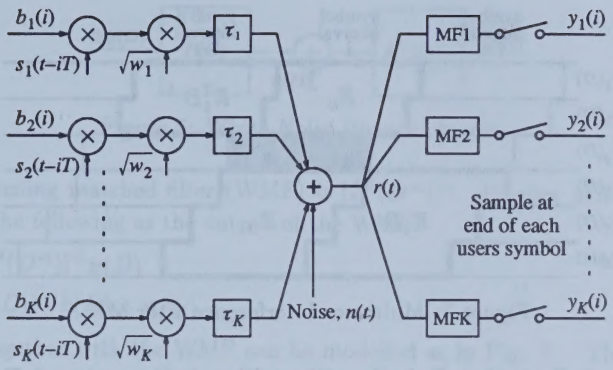


Figure 1: Asynchronous DS/SSMA System Model.

Now it is possible to write the output of the DTC in terms of the input as

$$\boldsymbol{y}(D) = \boldsymbol{S}(D)\boldsymbol{W}\boldsymbol{b}(D) + \boldsymbol{z}(D), \tag{1}$$

where $\boldsymbol{W}$ is a diagonal $K \times K$ matrix of received user amplitudes ($W_{k,k} = \sqrt{w_k}$), $\boldsymbol{S}(D)$ is the correlation spectrum for a particular set of user signature waveforms and delays $\{s_k, \tau_k\}_1^K$, and $\boldsymbol{z}(D)$ is a K element coloured Gaussian noise vector with spectrum $E[\boldsymbol{z}(D)\boldsymbol{z}^{\mathrm{T}}(D^{-1})] = N_0\boldsymbol{S}(D)$. The system in Fig. 1 is now equivalent to the discrete time channel shown in Fig. 2.

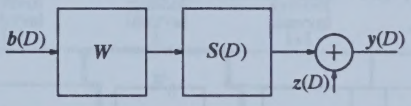


Figure 2: Discrete Time Channel Model.

$\boldsymbol{S}(D)$ can be written in terms of two $K \times K$ cross-correlation matrices as

$$\boldsymbol{S}(D) = \boldsymbol{R}_1^{\mathrm{T}}D^{-1} + \boldsymbol{R}_0 + \boldsymbol{R}_1 D. \tag{2}$$

$\boldsymbol{R}_1^{\mathrm{T}}D^{-1}$ represents energy collected from the next symbol interval, $\boldsymbol{R}_0$ represents energy collected from the current symbol interval, and $\boldsymbol{R}_1 D$ represents energy collected from the previous symbol interval. The scope of multiuser interference collected for a particular user $k = 3$ during a particular symbol interval i is visualised in Fig. 3 for a 6 user example.

We see that there are $2K - 1$ symbols that effect a particular $y_k(i)$. There can be less than $2K - 1$ if the overlapped sections of the signature waveforms are orthogonal. The elements of the cross-correlation matrices are

$$[\boldsymbol{R}_0]_{k,l} = \int_{-\infty}^{+\infty} s_k(t - \tau_k)\, s_l(t - \tau_l)dt, \text{ and}$$

$$[\boldsymbol{R}_1]_{k,l} = \int_{-\infty}^{+\infty} s_k(t - \tau_k)\, s_l(t + T - \tau_l)dt.$$

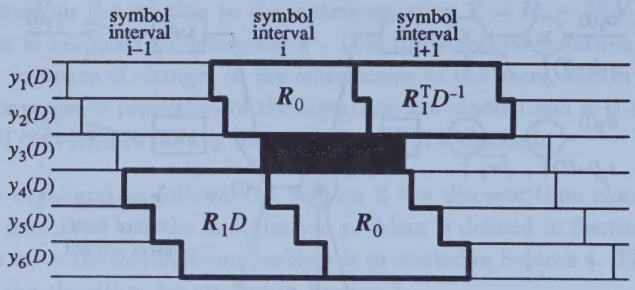


Figure 3: Multiuser Interference after MF.

These equations indicate that $\boldsymbol{R}_0$ is Hermitian with unit diagonal, and $\boldsymbol{R}_1$ is upper right triangular with zero diagonal.

3 Channel Factorisation

The channel of Eqn. 2 is represented by a matrix with elements that are polynomials in D. This matrix can be factored as

$$\boldsymbol{S}(D) = \boldsymbol{F}^{\mathsf{T}}(D^{-1})\boldsymbol{F}(D),$$

where $\boldsymbol{F}(D^{-1})$ and $\boldsymbol{F}(D)$ are causal matrix filters. Now we can represent the factored DTC as shown in Fig. 4.

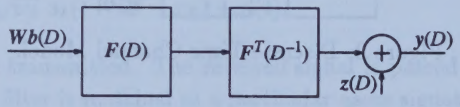


Figure 4: Factorised Channel Model.

It is possible to show that $\boldsymbol{F}(D) = \boldsymbol{F}_0 + \boldsymbol{F}_1 D$, where $\boldsymbol{F}_0$ is lower left triangular and $\boldsymbol{F}_1$ is upper right triangular with zero diagonal, is a sufficient structure to solve the decomposition of $\boldsymbol{S}(D)$. A simple substitution reveals that

$$\boldsymbol{R}_0 = \boldsymbol{F}_0^{\mathsf{T}}\boldsymbol{F}_0 + \boldsymbol{F}_1^{\mathsf{T}}\boldsymbol{F}_1, \text{ and} \tag{3}$$

$$\boldsymbol{R}_1 = \boldsymbol{F}_0^{\mathsf{T}}\boldsymbol{F}_1. \tag{4}$$

We can move the noise $\boldsymbol{z}(D)$ to a position between the two elements of the factored channel by altering its correlation spectrum [6]. The resulting channel model has white noise $\boldsymbol{n}(D), E[\boldsymbol{n}(D)\boldsymbol{n}^{\mathsf{T}}(D^{-1})] = N_0$ as shown in Fig. 5. Eqn. 1 can now be rewritten as

$$\begin{aligned}\boldsymbol{y}(D) &= \boldsymbol{S}(D)\boldsymbol{W}\boldsymbol{b}(D) + \boldsymbol{z}(D) \\ &= \boldsymbol{F}^{\mathsf{T}}(D^{-1})\left[\boldsymbol{F}(D)\boldsymbol{W}\boldsymbol{b}(D) + \boldsymbol{n}(D)\right]\end{aligned} \tag{5}$$

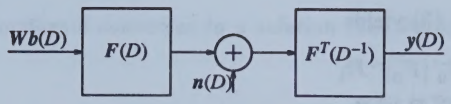


Figure 5: White Noise Channel Model.

The noise whitening matched filter (WMF) is $[\boldsymbol{F}^{\mathsf{T}}(D^{-1})]^{-1}$. Passing $\boldsymbol{y}(D)$ through this filter we have the following as the output of the WMF.

$$\begin{aligned}\boldsymbol{y}'(D) &= [\boldsymbol{F}^{\mathsf{T}}(D^{-1})]^{-1}\boldsymbol{y}(D) \\ &= \boldsymbol{F}(D)\boldsymbol{W}\boldsymbol{b}(D) + \boldsymbol{n}(D) \end{aligned} \tag{6}$$

The channel together with the WMF can be modelled as in Fig. 6. The noise is now

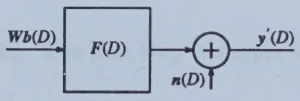


Figure 6: WMF and Channel Model.

white in the statistic $\boldsymbol{y}'(D)$. From a multiuser interference point of view we have also partially decorrelated the statistic. The scope of multiuser interference for a particular user k during a particular symbol interval i is visualised in Fig. 7 which can be compared with the matched filter output as shown in Fig. 3.

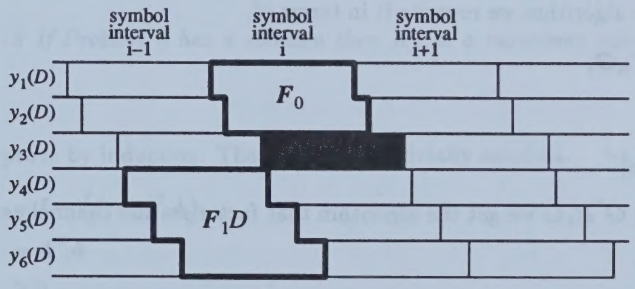


Figure 7: Multiuser Interference after WMF.

In many new sub–optimal tree search algorithms the use of the WMF improves the performance significantly. However, the determination of $\boldsymbol{F}(D)$ from $\boldsymbol{S}(D)$ is not straightforward.

4 Algorithm

As described in [6] the determination of $\boldsymbol{F}(D)$ from $\boldsymbol{S}(D)$ amounts to solving Eqns. (3) and (4) simultaneously. We start by eliminating $\boldsymbol{F}_1$. From (4) $\boldsymbol{F}_1 = [\boldsymbol{F}_0^{\mathsf{T}}]^{-1}\boldsymbol{R}_1$.

Substituting into Eqn. (3) yields

$$\begin{aligned} \boldsymbol{R}_0 &= \boldsymbol{F}_0^{\mathrm{T}}\boldsymbol{F}_0 + \boldsymbol{R}_1^{\mathrm{T}}\boldsymbol{F}_0^{-1}[\boldsymbol{F}_0^{\mathrm{T}}]^{-1}\boldsymbol{R}_1 \\ &= \boldsymbol{F}_0^{\mathrm{T}}\boldsymbol{F}_0 + \boldsymbol{R}_1^{\mathrm{T}}[\boldsymbol{F}_0^{\mathrm{T}}\boldsymbol{F}_0]^{-1}\boldsymbol{R}_1 \end{aligned}$$

Through a variable substitution $\boldsymbol{Y} = \boldsymbol{F}_0^{\mathrm{T}}\boldsymbol{F}_0$ we get

$$\boldsymbol{R}_0 = \boldsymbol{Y} + \boldsymbol{R}_1^{\mathrm{T}}\boldsymbol{Y}^{-1}\boldsymbol{R}_1 \tag{7}$$

Our objective is now to solve Eqn. (7) for $\boldsymbol{Y}$. Once we have $\boldsymbol{Y}$, Cholesky decomposition [7] will yield $\boldsymbol{F}_0$. Solving Eqn. (7) for $\boldsymbol{Y}$ amounts to solving $K^2/2$ non–linear equations for $K^2/2$ unknowns. We offer an alternative solution method. If Eqn. (7) has a solution we will prove that under certain restrictions the solution can be found via the following recursive algorithm:

Algorithm 1

$$\begin{aligned} \boldsymbol{X}_0 &= \boldsymbol{R}_0 \\ \boldsymbol{X}_{n+1} &= \boldsymbol{R}_0 - \boldsymbol{R}_1^{\mathrm{T}}\boldsymbol{X}_n^{-1}\boldsymbol{R}_1 \end{aligned}$$

The complexity of the algorithm is $O(K^3/2)$ operations per iteration. Notice that this is less than the complexity of matrix inversion or matrix multiplication which are $O(K^3)$. This saving is due to the symmetry of the equation system.

To analyse the algorithm we re-write it in terms of

$$\boldsymbol{X}_n \triangleq \boldsymbol{G}^{\mathrm{T}}\boldsymbol{Y}_n\boldsymbol{G},$$

where

$$\boldsymbol{G}\boldsymbol{G}^{\mathrm{T}} \triangleq [\boldsymbol{R}_0]^{-1}.$$

so letting $\boldsymbol{A} = \boldsymbol{G}^{\mathrm{T}}\boldsymbol{R}_1\boldsymbol{G}$ we get the algorithm that factorises the channel as

Algorithm 2

$$\begin{aligned} \boldsymbol{X}_0 &= \boldsymbol{I} \\ \boldsymbol{X}_{n+1} &= \boldsymbol{I} - \boldsymbol{A}^{\mathrm{T}}\boldsymbol{X}_n^{-1}\boldsymbol{A} \end{aligned}$$

Algorithm 2 is easier to work with that Algorithm 1 even though they both solve problem 7. Moreover, Algorithm 2 solves

$$\boldsymbol{X} = \boldsymbol{I} - \boldsymbol{A}^{\mathrm{T}}\boldsymbol{X}^{-1}\boldsymbol{A}, \tag{8}$$

which is equivalent to Problem 7.

In the following Lemmas, $\boldsymbol{A} > 0$ denotes that $\boldsymbol{A}$ is a positive definite matrix, $\boldsymbol{A} \geq 0$ denotes that $\boldsymbol{A}$ is a positive semi–definite matrix, and $\boldsymbol{A} > \boldsymbol{B} \Leftrightarrow \boldsymbol{A} - \boldsymbol{B} > 0$ denotes that $\boldsymbol{A} - \boldsymbol{B}$ is a positive definite matrix.

Theorem 4.1 *If Algorithm 2 converges to a solution then Problem 8 has the same solution.*

Proof: Algorithm 2 has found a solution that converges to $\boldsymbol{X}_n$. So

$$\begin{aligned}\boldsymbol{X}_n &= \boldsymbol{I} - \boldsymbol{A}^{\mathsf{T}}[\boldsymbol{X}_{n-1}]^{-1}\boldsymbol{A} \\ &= \boldsymbol{I} - \boldsymbol{A}^{\mathsf{T}}[\boldsymbol{X}_n]^{-1}\boldsymbol{A}.\end{aligned}$$

By setting $\boldsymbol{X} = \boldsymbol{X}_n$ we have a solution to $\boldsymbol{X} = \boldsymbol{I} - \boldsymbol{A}^{\mathsf{T}}\boldsymbol{X}^{-1}\boldsymbol{A}$ and therefore solved Problem 8. This completes the proof. □

Theorem 4.2 *If Problem 8 has a solution then* $\boldsymbol{X}_n \geq \boldsymbol{X}$ $\forall n$.

Proof: We prove by induction. The first step is simply satisfied since $0 < \boldsymbol{X} \leq \boldsymbol{I}$.

$$\boldsymbol{X}_0 - \boldsymbol{X} = \boldsymbol{I} - \boldsymbol{X}.$$

Now assume that $\boldsymbol{X}_n - \boldsymbol{X} \geq 0$. Then, we have $[\boldsymbol{X}_n]^{-1} \leq \boldsymbol{X}^{-1}$ so

$$\begin{aligned}\boldsymbol{X}_{n+1} - \boldsymbol{X} &= \boldsymbol{A}^{\mathsf{T}}\left[\boldsymbol{X}^{-1} - [\boldsymbol{X}_n]^{-1}\right]\boldsymbol{A} \\ &\geq 0.\end{aligned}$$

This completes the proof. □

Theorem 4.3 *If Problem 8 has a solution then* $\boldsymbol{X}_n$ *is a monotone non-increasing sequence in* n.

Proof: We prove by induction. The first step is trivially satisfied.

$$\begin{aligned}\boldsymbol{X}_0 - \boldsymbol{X}_1 &= \boldsymbol{I} - \left(\boldsymbol{I} - \boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\right) \\ &= \boldsymbol{A}^{\mathsf{T}}\boldsymbol{A} \\ &\geq 0.\end{aligned}$$

Now we assume

$$\begin{aligned}&\boldsymbol{X}_j \geq \boldsymbol{X}_{j+1} \\ \Leftrightarrow\ &[\boldsymbol{X}_j]^{-1} \leq [\boldsymbol{X}_{j+1}]^{-1}.\end{aligned}$$

Directly from the recursive expressions we have

$$\begin{aligned}\boldsymbol{X}_{j+1} - \boldsymbol{X}_{j+2} &= \boldsymbol{A}^{\mathsf{T}}\left([\boldsymbol{X}_{j+1}]^{-1} - [\boldsymbol{X}_j]^{-1}\right)\boldsymbol{A} \\ &\geq 0.\end{aligned}$$

This induction is therefore complete and the theorem proved. □

Theorem 4.4 *If Problem 8 has a solution then Algorithm 2 will find the same solution.*

Proof: Since Problem 8 has a solution we have the following:

1. $\boldsymbol{X}_n$ is bounded below by $\boldsymbol{X}$ by Theorem 4.2, and
2. $\boldsymbol{X}_n$ is monotonically non-increasing in n by Theorem 4.3.

We therefore conclude that $\boldsymbol{X}_n$ will converge to $\boldsymbol{X}$ as n increases. This completes the proof. □

5 Sufficient Conditions for Convergence

In this section we will study the conditions under which Algorithm 2 will find a solution to Problem 8. We begin with the following definition,

$$\mathcal{R} = \begin{bmatrix} \boldsymbol{R}_0 & \boldsymbol{R}_1^{\mathsf{T}} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{R}_1 & \boldsymbol{R}_0 & \boldsymbol{R}_1^{\mathsf{T}} & & \vdots \\ \boldsymbol{0} & \boldsymbol{R}_1 & \boldsymbol{R}_0 & \ddots & \boldsymbol{0} \\ \vdots & & \ddots & \ddots & \boldsymbol{R}_1^{\mathsf{T}} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{R}_1 & \boldsymbol{R}_0 \end{bmatrix}$$

where $\mathcal{R}$ is a $MK \times MK$ matrix. To show the conditions for the convergence of Algorithm 1 we begin with the realisation that $\boldsymbol{X}_n$ is found from the Cholesky decomposition of $\mathcal{R}$ with $M = n$. We start with the following lemma.

Lemma 5.1 *If $\mathcal{R} > 0$ then Problem 8 has a solution.*

Proof: Solving Problem 8 is equivalent to solving $\mathcal{R} = \mathcal{F}^{\mathsf{T}}\mathcal{F}$ for $\mathcal{F}$, where $\mathcal{F}$ is a lower left triangular matrix with non–zero diagonal. From Cholesky decomposition [7] we know that if $\mathcal{R} > 0$ then we can find $\mathcal{F}$. This completes the proof. □

It remains to show that $\mathcal{R}$ is positive definite, thus ensuring the applicablity of our algorithm.

Lemma 5.2 *$\mathcal{R}$ is positive semi–definite, and almost surely (in a mathematical sense) positive definite.*

Proof: Let us assume that the delay of each users spreading waveform at the receiver is quantised to the nearest T_d seconds. The spreading waveform can be demodulated to yield the spreading code. If each chip of the spreading code is of length T_c seconds then the number of possible values of τ_k that lie within one chip interval is

$$L = \frac{T_c}{T_d}.$$

We assume that T_d is a multiple of T_c. The number of chips per spreading code is N. τ_k can therefore take only NL possible values distributed uniformly in $[0, T)$. So

$$\tau_k = d_k T_d,$$

where d_k is an integer $\in [0, NL-1]$. We can now place the received spreading codes into the $2NL \times K$ matrix C where

$$[C]_{n,k} = \begin{cases} \frac{1}{\sqrt{NL}} \boldsymbol{code}_k((n-d_k)/L) & d_k \le n \le N + d_k \\ 0 & \text{otherwise.} \end{cases}$$

Here $\boldsymbol{code}_k(i)$ is the ith chip $\in \{-1, 1\}$ for user k. Now let us partition $\boldsymbol{C}$ between rows NL and $NL+1$ so that

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{C}_0 \\ \boldsymbol{C}_1 \end{bmatrix}.$$

This partitioning means that $\boldsymbol{C}_0$ and $\boldsymbol{C}_1$ are both $NL \times K$ matrices. We can write the correlation matrices for the spreading waveforms, assuming square chip waveforms as follows:

$$\boldsymbol{R}_0 = \boldsymbol{C}_0^{\mathsf{T}} \boldsymbol{C}_0 + \boldsymbol{C}_1^{\mathsf{T}} \boldsymbol{C}_1$$
$$\boldsymbol{R}_1 = \boldsymbol{C}_0^{\mathsf{T}} \boldsymbol{C}_1.$$

We note that we can write $\boldsymbol{\mathcal{R}} = \boldsymbol{\mathcal{C}}^{\mathsf{T}} \boldsymbol{\mathcal{C}}$, where

$$\boldsymbol{\mathcal{C}} = \begin{bmatrix} \boldsymbol{C}_0 & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{C}_1 & \boldsymbol{C}_0 & \boldsymbol{0} & & \vdots \\ \boldsymbol{0} & \boldsymbol{C}_1 & \boldsymbol{C}_0 & \ddots & \boldsymbol{0} \\ \vdots & & \ddots & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{C}_1 & \boldsymbol{C}_0 \end{bmatrix}$$

is an $MNL \times MKL$ matrix. This formulation of $\boldsymbol{\mathcal{R}}$ proves the positive *semi-definiteness* of $\boldsymbol{\mathcal{R}}$. For $\boldsymbol{\mathcal{R}}$ to be positive semi-definite, the received spreading waveforms must be linearly dependent. Lupas and Verdú have devised sufficient conditions for the linear independence assumption in [8]. If the time delays $\boldsymbol{\tau} = \{\tau_1, \tau_2, \ldots, \tau_K\}$ are independent, continuous random variables, then the received spreading waveforms are, in a mathematical sense, almost surely linearly independent. Details can be found in [8]. It then follows that as $T_d \to 0$ the matrix $\boldsymbol{\mathcal{R}} > 0$, which completes the proof. □

We have seen that $\boldsymbol{\mathcal{R}}$ is nearly always positive definite and therefore a solution to Problem 8 nearly always exists. In Section 4 we showed that if Problem 8 has a solution then Algorithm 2 will find it.

6 Summary

We have proposed a technique to factorise the time–invariant multiuser asynchronous CDMA channel in an efficient manner. The technique is recursive and is therefore adaptive to continuous changes in the correlation of the users spreading codes. A constraint on the correlations of the spreading codes to ensure convergence was derived. This constraint was shown to be consistent with nearly all possible non–singular CDMA channels.

7 Acknowledgements

The authors thank Telecom Australia and the Commonwealth of Australia for the financial support of this work.

References

[1] S. Verdú, "Optimum multiuser asymptotic efficiency," *IEEE Transactions on Communications*, vol. 34, pp. 85–96, Sept. 1986.

[2] P. D. Alexander, "Reduced complexity detectors for asynchronous DS/SSMA," in *IEEE International Conference on Universal Wireless Access*, pp. 7–11, Apr. 1994.

[3] A. Duel-Hallen, "Decorrelating decision–feedback multiuser detector for synchronous code–division multiple–access channel," *IEEE Transactions on Communications*, vol. 41, pp. 285–290, February 1993.

[4] Z. Xie, C. K. Rushforth, R. T. Short, and T. K. Moon, "Joint signal detection and parameter estimation in multiuser communications," *IEEE Transactions on Communications*, vol. 41, pp. 1208–1216, Aug. 1993.

[5] A. Duel-Hallen, "A family of multiuser detection–feedback detectors for asynchronous code–division multiple–access channels," *IEEE Transactions on Communications, to appear.*

[6] P. D. Alexander and L. K. Rasmussen, "An efficient technique for deriving receiver filters in multiuser asynchronous DS/SSMA," in *The Fifth International Symposium on Personal, Indoor and Mobile Radio Communications*, 1994.

[7] G. Golub and C. V. Loan, *Matrix Computations.* The John Hopkins University Press, 1983.

[8] R. Lupas and S. Verdú, "Near-far resistance of multiuser detectors in asynchronous channels," *IEEE Transactions on Communications*, vol. 38, pp. 496–508, Apr. 1990.

A Simple Way to Compute the Minimum Distance in Multiuser CDMA Systems*

Christian Schlegel and Lei Wei

Christian Schlegel, Division of Engineering, University of Texas at San Antonio,
6900 North Loop, 1604 west San Antonio, TX 78249 USA
Lei Wei, Signal Processing Research Institute, University of South Australia,
The Levels, SA5095, Australia,

* This work was funded by Telecom Australia and the Commonwealth of Australia.

Abstract: A simple method to compute the minimum distance in multiuser CDMA systems based on the output of the Whitening Matched filter is proposed. In contrast to the results in [2][3], which are based on the output of the Matched filter, the computation of the asymptotic efficiency of the optimum multiuser detection is not an NP–hard problem. Numerical results are presented for synchronous CDMA with binary random signature waveforms of length 31.

I INTRODUCTION

Recently, joint multiuser detection, in which the multiuser interference is treated as a part of the information, rather than noise, has attracted much attention [1]–[9]. The work of Verdú and Lupas has shown that optimum near–far resistance and a significant performance improvement over the conventional detector can be achieved [1]–[4]. The substantial improvements, however, are obtained at the expense of a dramatic increase in computational complexity. The complexity grows exponentially with the number of users. Thus, when the number of users is large, the optimum detector becomes infeasible. It is therefore desirable to use a near optimum, low complexity detector for CDMA systems with a large number of users.

Since then, many low complexity multiuser detectors have been proposed. Lupas and Verdú considered a linear multiuser detector in [3] [4]. The linear multiuser detector achieves optimum near–far resistance but cannot provide close to optimum performance. A multistage technique was proposed by Varanasi and Aazhang [5] [6], and Duel–Hallen suggested the decorrelating decision feed–back detector (DDFD) [7]. Due to error propagation, the multistage multiuser detector and the DDFD achieve close–to–optimum performance *only* when the interfering users are significantly stronger than the user under consideration (i.e. the weakest user benefits the most). In Wei and Schlegel's improved DDFD (IDDFD) [8] [9] the hard decision feed–back of the DDFD is replaced by soft decision feed–back, which effectively eliminates error propagation and achieves close–to–optimum performance for all users.

Many suboptimum detectors have been claimed to be near optimum. However, a reliable performance measure like the bit error rate or the asymptotic efficiency of optimum detection is not known for large numbers of users, except for the single user bound which is a simple performance upper bound. This makes it hard to evaluate suboptimum detectors.

The performance and the asymptotic efficiency of optimum multiuser detection have been studied by Verdu [1] [2]. The computation of the asymptotic efficiency of optimum multiuser

detection has been proved to be an NP–hard problem [2]. However, Verdu's work, which like most early work on multiuser detection, is based on the Matched filter only. As we have already noted, suboptimum detectors can achieve improved performance if they are based on the Whitening Matched filter rather than the Matched filter only [7]–[10]. In this paper we present a simple way to compute the minimum distance of the optimum multiuser detection which is also based on the output of the Whitening Matched filter. That is, we demonstrate that the computation of the asymptotic efficiency of the optimum multiuser detection is not an NP–hard problem.

In this paper we will concentrate on synchronous CDMA, however the method presented here can be generalized for asynchronous CDMA, for multiuser detection with error control coding and for ISI channels.

In Section II, we present the signal and system of synchronous CDMA with multiuser detection. In Section III, we give the simple method to calculate the minimum distance of optimum multiuser detection. In Section IV we show numerical results, and, in Section V, the discussion and conclusions are given.

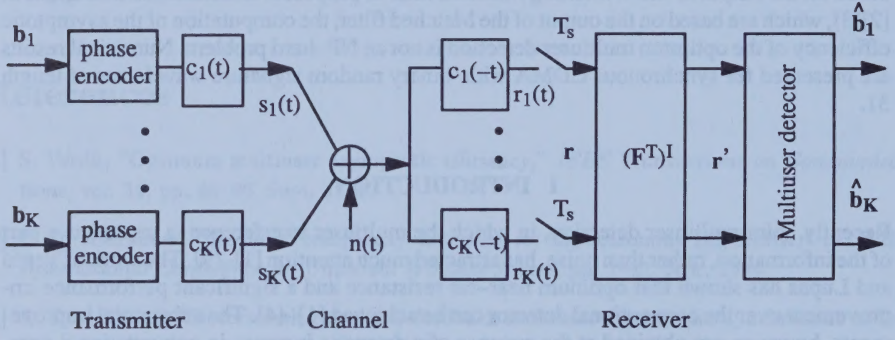


FIGURE 1 –Baseband equivalent model of system

II SIGNAL AND SYSTEM

Consider a synchronous CDMA system with K users and a set of unity energy preassigned signature waveforms ($s_k(t)$, k=1,2,..., K) of duration T_s as shown in Figure 1. At the receiver, a bank of filters matched to the set of preassigned waveforms is sampled at times ℓT_s.

The sample values

$$\mathbf{r} = [r_1(\ell T), ..., r_K(\ell T)]^T = [r_1, ..., r_K]^T = \mathbf{RWb} + \mathbf{z} \ , \qquad (1)$$

form a sufficient statistic for optimum multiuser detection [7], where $\mathbf{b}=[b_1, b_2, ..., b_K]$, and $\{b_k \in \{-1,1\}\}$ is the k–th user information sequence, $\mathbf{R}$ is the nonnegative definite correlation matrix, $R_{ij}=\int_0^{T_s} s_i(t)s_j(t)dt$, and $\mathbf{W}= \mathrm{diag}\left[\sqrt{w_1}, \sqrt{w_2}, ..., \sqrt{w_K}\right]$, where w_k is the energy–per–bit. The superscript T denotes the matrix transposition. The noise vector $\mathbf{z}$ in equation (1) is a Gaussian noise vector of dimension K with the $K \times K$ autocorrelation matrix $R(\mathbf{z})=N_o\mathbf{R}/2$, where N_o is the one–sided noise power spectral density of a zero–mean additive

white Gaussian noise (AWGN) source and $R(\mathbf{z})_{k,j}=E[z_k, z_j]$.
The preassigned signature waveforms are based on discrete signature sequences. In a synchronous CDMA system the relationship between the correlation matrix $\mathbf{R}$ and the $K \times N$ matrix of discrete signature sequences $\mathbf{C}$ is given by $\mathbf{R} = \mathbf{C}^T\mathbf{C}$, where

$$\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_K] = \begin{bmatrix} c_{1,1} & c_{2,1} & \cdots & c_{K,1} \\ c_{1,2} & c_{2,2} & \cdots & c_{K,2} \\ \cdots & \cdots & \cdots & \cdots \\ c_{1,N} & c_{2,N} & \cdots & c_{K,N} \end{bmatrix}, \tag{2}$$

$\mathbf{c}_k = [c_{k,1}, c_{k,2}, ..., c_{k,N}]^T$ is the discrete signature sequence of the k–th user and $c_{i,j} \in \{-1, 1\}$ for binary signature sequences.
If $\mathbf{R}$ is a positive definite matrix (non–singular) there exists a unique lower triangular, non–singular matrix $\mathbf{F}$ such that $\mathbf{R}=\mathbf{F}^T\mathbf{F}$.
Let $\mathbf{X}^I$ denote the inverse of a matrix $\mathbf{X}$. When the whitening filter $(\mathbf{F}^T)^I$ is applied to the sampled outputs of the matched filter (Equation (1)), the output vector is [7]

$$\mathbf{r}' = [r_1', r_2', ..., r_K']^T = \mathbf{FWb} + \mathbf{n}, \tag{3}$$

where $\mathbf{n}$ is a white Gaussian noise vector with autocorrelation matrix $R(\mathbf{n})=N_o\mathbf{I}_K/2$, and $\mathbf{I}_K$ is the $K \times K$ identity matrix. The decision rule for the optimum multiuser detector is to select the symbol vector $\hat{\mathbf{b}}$ which minimizes the Euclidean metric $\|\mathbf{r}'-\mathbf{FWb}\|^2$ [8], or maximizes

$$\hat{\mathbf{b}} \in \arg \max_{\mathbf{b} \in \{-1,1\}^K} \left(-\|\mathbf{r}'-\mathbf{FWb}\|^2\right) = \arg \max_{\mathbf{b} \in \{-1,1\}^K} \sum_{k=1}^{K} -m_k . \tag{4}$$

where

$$m_k = \left[r_k' - \sum_{j=1}^{k} F_{kj}\sqrt{w_j}\, b_j \right]^2 \tag{5}$$

is the *k*–th metric function based on the whitening filter outputs.

III THE MINIMUM DISTANCE CALCULATION

Clearly, the minimum distance of user #i of the optimum multiuser detection for synchronous CDMA is

$$D_{min}(i) = \min_{\substack{\mathbf{d}, \mathbf{b} \in \{-1,1\}^K \\ d_i \neq b_i}} \|\mathbf{FWd}-\mathbf{FWb}\|^2 . \tag{6}$$

Let $e_j=(d_j-b_j)/2$ for the purpose of normalization for j=1, 2, ..., K. Then, Equation (6) can be rewritten as

$$D_{min,norm}(i) = D_{min}(i)/4 = \min_{\substack{\mathbf{e} \in \{-1, 0, 1\}^K \\ e_i \neq 0}} \sum_{k=1}^{K} D_k , \tag{7}$$

where $\mathbf{e}=[e_1, e_2, ..., e_K]$,

$$D_k = \left[\sum_{j=1}^{k} F_{kj} \sqrt{w_j} e_j \right]^2 . \qquad (8)$$

Since $D_k \geq 0$ and $D_{min,norm}(i) \leq w_i/4$ (from the single user bound), we have the following simple procedure to find the minimum distance of user i, using the definition $\mathfrak{D}_k = D_1 + D_2 + ... + D_k$:

1) Let k=1
2) Compute $\mathfrak{D}_k = \mathfrak{D}_{k-1} + D_k$ for all survived paths, that is the paths with $\mathfrak{D}_{k-1} < w_i$.
3) Drop the paths with $\mathfrak{D}_k \geq w_i$.
4) k=k+1; If k=K stop, otherwise go to 2).

Since the paths with $\mathfrak{D}_k \geq w_i$ are dropped, the number of tree branches searched by the above procedure is significantly smaller than that of the the full tree (3^{K-1}). If the search is based on the Matched filter output [1,2], paths with $\mathfrak{D}_{k-1} \geq w_i$ can not be dropped since D_k can be negative.

IV NUMERICAL RESULTS

In this section we use the computation simulation to evaluate the minimum distance of synchronous CDMA with binary random signature waveforms of length 31. There are several reasons for considering binary random signature waveforms. Firstly, random waveforms have been used in practical systems such as QualComm CDMA [11]. Secondly, the performance of a CDMA system with random codes can be viewed as an upper bound on the BEP of a system, since we can only improve on the performance of a system with random codes by selecting a 'good' set of signature waveforms.

The simulation procedure is as follows. First, a set of random signature waveforms of length 31 is generated for every bit duration T and the correlation matrix **R** is computed. Then we check whether the matrix **R** is singular. If $F_{kk}=0$, then **R** is singular and the whitening filter does not exist, hence if $\min[F_{kk},\ k = 1,..,K] = 0$, the set is discarded and a new set of random waveforms is generated. The signature waveforms with $\min[F_{kk},\ k = 1,...,K] \geq 0$ are used to spread the signal at the transmitter and to detect the signal at the receiver. A total of 100 sets of random signature waveforms were examined and none of the sets had to be discarded. In this section let $W_k/4=1$ for k=1, 2, ..., K.

In Figure 2, we show the widths of the surviving tree as a function of the user index for a set of random signature waveforms. (Among the first five sets generated the one which had the maximum width was chosen). The maximum width of the surviving tree is 48479 which is significantly less than the maximum number of tree branches ($3^{30}=2.1 \times 10^{14}$ for K=31).

In Figure 3 and 4, we show a histogram of the minimum distances of the binary random signature waveform sets which have been tested. Interestingly, we found that the 100 sets of random waveforms have only 5 discrete values of the minimum distance. These values are 0.516, 0.645, 0.774, 0.903, 1.0. Figure 3 shows that the minimum distances for a full set of users, while Figure 4 shows the case for a smaller number of users. The minimum distances in Figure 4 concentrate much more around the value 1.0. It seems unreasonable to assume that the above values for the minimum distance are the only once that exist, but any other values appear to be very unlikely.

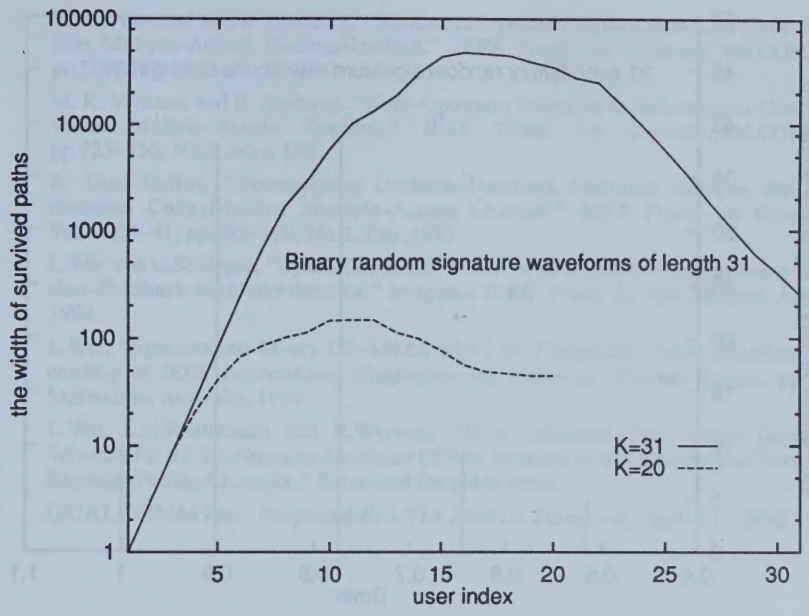


FIGURE 2.The widths of the survived tree branches as a function of the user index.

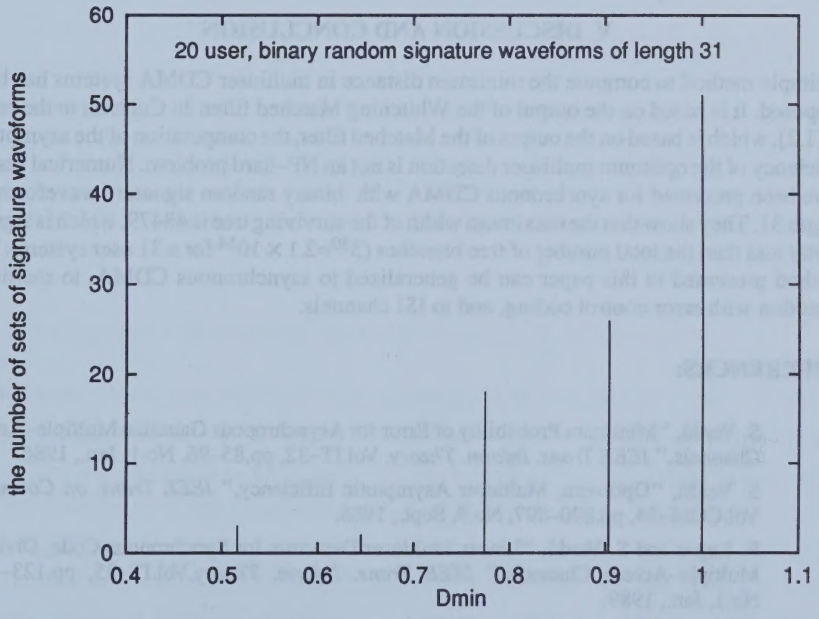


FIGURE 3. Minimum distances for a 31 user system.

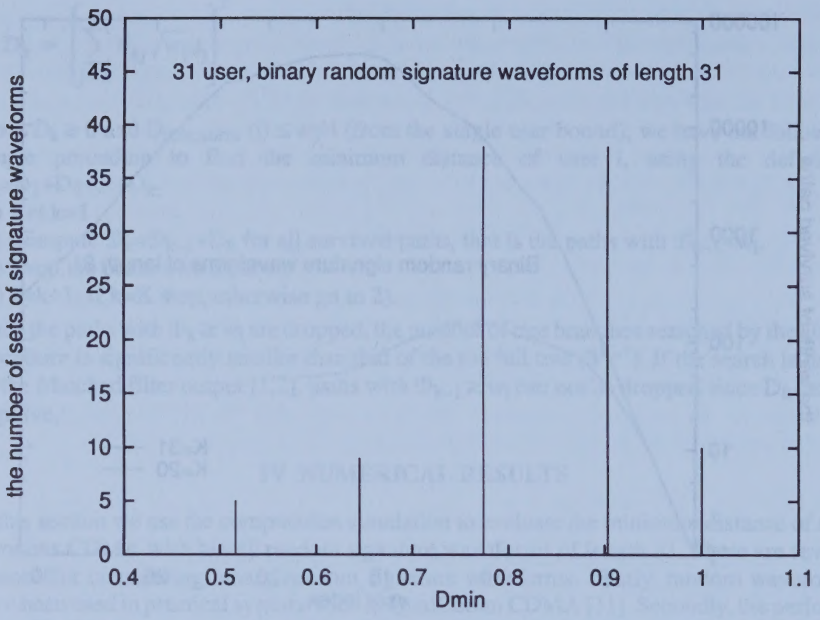


FIGURE 4. Minimum distance for a 20 user system.

V DISCUSSION AND CONCLUSION

A simple method to compute the minimum distance in multiuser CDMA systems has been proposed. It is based on the output of the Whitening Matched filter. In Contrast to the result of [1,2], which is based on the output of the Matched filter, the computation of the asymptotic efficiency of the optimum multiuser detection is not an NP–hard problem. Numerical results have been presented for synchronous CDMA with binary random signature waveforms of length 31. They show that the maximum width of the surviving tree is 48479, which is significantly less than the total number of tree branches ($3^{30}=2.1 \times 10^{14}$ for a 31 user system). The method presented in this paper can be generalized to asynchronous CDMA, to multiuser detection with error control coding, and to ISI channels.

REFERENCES:

[1] S. Verdú, "Minimum Probability of Error for Asynchronous Gaussian Multiple–Access Channels," *IEEE Trans. Inform. Theory,* Vol.IT–32, pp.85–96, No.1, Jan., 1986.

[2] S. Verdú, "Optimum, Multiuser Asymptotic Efficiency," *IEEE Trans. on Commun.,* Vol.COM–34, pp.890–897, No.9, Sept., 1986.

[3] R. Lupas and S. Verdú, "Linear Multiuser Detectors for Synchronous Code_Division Multiple–Access Channels," *IEEE Trans. Inform. Theory,*Vol.IT–35, pp.123–136, No.1, Jan., 1989.

[4] R. Lupas and S. Verdú, "Near–Far Resistance of Multiuser detectors in Asynchronous Channels," *IEEE Trans. on Commun.,* Vol.COM–38, pp.496–508, No.4, April, 1990.

[5] M. K. Varanasi and B. Aazhang, "Multistage Detection in Asynchronous Code–Division Muliple–Access Communications," *IEEE Trans. on Commun.*,Vol.COM–38, pp.509–519, No.4, April, 1990.

[6] M. K. Varanasi and B. Aazhang, "Near–Optimum Detection in Synchronous Code–Division Muliple–Access Systems," *IEEE Trans. on Commun.*,Vol.COM–39, pp.725–736, No.5, May, 1991.

[7] A. Duel–Hallen, "Decorrelating Decision–Feedback Multiuser Detector for Synchronous Code–Division Multiple–Access Channel," *IEEE Trans. on Commun.*, Vol.COM–41, pp.285–290, No.2, Feb.,1993

[8] L.Wei and C.Schlegel, "Synchronous DS–SSMA with Improved Decorrelating Decision–Feedback Multiuser detector," to appear *IEEE Trans. on Veh. Technol. August, 1994.*

[9] L.Wei, "Synchronous M–ary DS–SSMA with Low Complexity Joint Detection,'Proceeding of *IEEE International Conference on Universal Wireless Access*, pp.1–5, Melbourne, Australia, 1994.

[10] L.Wei, L.K.Rasmussen and R.Wyrwas, "Near Optimum Tree–Search Detection Schemes for Bit Synchronous Multiuser CDMA Systems Over Gaussian and Two–path Rayleigh Fading Channels," Submitted for publication.

[11] QUALCOMM Inc., *Proposed EIA/TIA Interim Standard*, April 21, 1992.

SOFT DECISION ADAPTIVE MULTIUSER CDMA DETECTOR FOR ASYNCHRONOUS AWGN CHANNELS

Frank Viehofer **Zoran Siveski** **Yeheskel Bar-Ness**
Center for Communications and Signal Processing
Department of Electrical and Computer Engineering
New Jersey Institute of Technology
University Heights
Newark, NJ 07102 USA

Abstract

A multiuser detector in an additive white Gaussian noise, code-division multiple-access (CDMA) channel is proposed and analyzed. It employs a combination of a decorrelator and a nonlinear multiuser interference canceler that utilizes soft tentative decisions. The weights of the canceler are adaptively adjusted, in a manner that renders the knowledge of received signal energies and the use of training sequences unnecessary. The steady state error performance of the detector is obtained and found to be superior to the performance of the same detector that utilizes hard tentative decisions.

1 Preliminaries

In a multiuser environment, K users share the same channel with the unit energy signature waveform $s_k(t)$, $k = 1, \ldots, K$, of duration T assigned to each of them. The information bits $b_k(i) \in \{\pm 1\}$ for the symbol interval i have the same duration T. The waveform $r(t)$ at the input of the detector, which has a bank of matched filters as its front end, is expressed as:

$$r(t) = \sum_{k=1}^{K} \sum_{i} b_k(i) \sqrt{a_k} s_k(t - iT - \tau_k) + n(t),$$

where $n(t)$ is a zero-mean, white Gaussian noise with the two-sided power spectral density $N_0/2$, and a_k and τ_k are the received energy and relative delay for user k, respectively. While it is assumed that precise relative delay estimates are available for all users, their amplitudes are considered to be unknown to the detector.

Without a loss of generality, the attention will be on the detection of bit 0 of user 1, and it will be assumed that $0 = \tau_1 < \tau_2 < \ldots < \tau_K < T$. The sampled output of the matched filter for user 1 is then:

$$x_1(0) = \sqrt{a_1} b_1(0) + \sum_{k=2}^{K} \sqrt{a_k} \left[\rho_{k1} b_k(-1) + \rho_{1k} b_k(0) \right] + n_1(0).$$

The normalized partial cross-correlations for $k = 2, \ldots, K$ are:

$$\rho_{k1} = \int_0^T s_1(t)s_k(t + T - \tau_k)dt \quad \text{and} \quad \rho_{1k} = \int_0^T s_1(t)s_k(t - \tau_k)dt.$$

Also, $n_1(0) = \int_0^T n(t)s_1(t)dt$ is a zero-mean Gaussian random variable with variance $N_0/2$. Using the vector notations where $\boldsymbol{\rho}_1 = [\rho_{21}, \ldots, \rho_{K1}, \rho_{12}, \ldots, \rho_{1K}]^T$, $\boldsymbol{b}_1(0) = [b_2(-1), \ldots, b_K(-1), b_2(0), \ldots, b_K(0)]^T$, $\boldsymbol{A}_1 = diag[\sqrt{a_2}, \ldots, \sqrt{a_K}]$, and $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_1 \end{bmatrix}$, the matched filter output is:

$$x_1(0) = \sqrt{a_1}b_1(0) + \boldsymbol{\rho}_1^T \boldsymbol{A}\boldsymbol{b}_1(0) + n_1(0). \tag{1}$$

The multistage detector forms an estimate of the multiuser interference as the weighted vector of tentative decisions on symbols that interfere with $b_1(0)$ directly, and subtracts it from the matched filter output. The final decision statistic $y_1(0)$ and the corresponding final decision for bit 0 of user 1 are:

$$y_1(0) = x_1(0) - \boldsymbol{w}_1^T(0)\tilde{\boldsymbol{b}}_1(0) \qquad \text{and} \qquad \hat{b}_1(0) = \text{sgn}(y_1(0)), \tag{2}$$

where $\tilde{\boldsymbol{b}}_1(0) = [\tilde{b}_2(-1), \ldots, \tilde{b}_K(-1), \tilde{b}_2(0), \ldots, \tilde{b}_K(0)]^T$ is the vector of soft tentative decisions affecting bit 0 of user 1, and $\boldsymbol{w}_1(0) = [w_{21}^{(-1)}, \ldots, w_{K1}^{(-1)}, w_{21}^{(0)}, \ldots, w_{K1}^{(0)}]^T$ are the corresponding weights.

A detector that utilizes hard tentative decisions and fixed weights that require prior estimation of the received signals' amplitudes was first proposed in [1]. Several classes of similar multiuser detectors that use soft-decision tentative statistics were proposed and analyzed in [2]. A multistage detector using hard tentative decisions that operates without the knowledge of the received signals' amplitudes, and without the need to perform their prior estimation in order to set the values of the weights $\boldsymbol{w}_1(0)$ was presented in [3]. Instead, the weights are determined adaptively, without the requirement of a training sequence. The adaptive detector proposed here utilizes soft tentative decisions.

2 Proposed Multistage Adaptive Detector

In this paper, 1-shot decorrelated soft tentative decision statistics are considered. This decorrelator, described in [4] for a two-user case, can be extended in a straightforward manner for an arbitrary K. The vector of the decorrelator outputs $\boldsymbol{z}_1(0)$ and the corresponding tentative decisions affecting bit 0 of user 1 $\tilde{\boldsymbol{b}}_1(0)$ are:

$$\boldsymbol{z}_1(0) = \boldsymbol{A}\boldsymbol{b}_1(0) + \boldsymbol{\xi}_1 \qquad \text{and} \qquad \tilde{\boldsymbol{b}}_1(0) = \boldsymbol{g}(\boldsymbol{z}_1(0)), \tag{3}$$

where $\boldsymbol{\xi}_1 = [\xi_2(-1), \ldots, \xi_K(-1), \xi_2(0), \ldots, \xi_K(0)]^T$ is a zero-mean Gaussian vector having the covariance matrix $\boldsymbol{\Xi}_1$, whose diagonal elements are denoted by $\sigma_{\xi_k}^2$, and $\boldsymbol{g}(\cdot)$

denotes the tentative decision nonlinearity. Two different nonlinearities will be considered here; a dead zone limiter and a linear clipper. For a dead-zone limiter the elements of $\tilde{\boldsymbol{b}}_1(0)$ are defined as follows:

$$\tilde{b}_k(i) = \begin{cases} 0 & |z_k(i)| < \lambda_{k1}^{(i)} \\ \text{sgn}(z_k(i)) & \text{otherwise,} \end{cases} \tag{4}$$

where $i = -1, 0$ and $k = 2, 3, \ldots, K$.
In case of a linear clipper as a tentative decision nonlinearity:

$$\tilde{b}_k(i) = \begin{cases} z_k(i)/\lambda_{k1}^{(i)} & |z_k(i)| < \lambda_{k1}^{(i)} \\ \text{sgn}(z_k(i)) & \text{otherwise.} \end{cases} \tag{5}$$

The parameter $\lambda_{k1}^{(i)}$ is determined heuristically from the observed values of the decorrelator outputs as:

$$\lambda_{k1}^{(-1)} = \frac{\rho_{1k}^2[E\{|z_1(-1)|\}]^2}{E\{|z_k(-1)|\}} \quad \text{and} \quad \lambda_{k1}^{(0)} = \frac{\rho_{k1}^2[E\{|z_1(0)|\}]^2}{E\{|z_k(0)|\}}, \tag{6}$$

where the above expectations are evaluated as:

$$E\{|z_k(i)|\} = \sqrt{a_k}\left(1 - 2Q\left(\frac{\sqrt{a_k}}{\sigma_{\xi_k(i)}}\right)\right) + \frac{2\xi_k(i)}{\sqrt{2\pi}} \exp\left(-\frac{a_k}{2\sigma_{\xi_k(i)}^2},\right).$$

where

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-t^2/2} dt.$$

For controlling the weights, a steepest descent algorithm which minimizes the output signal energy $E\{y_1^2(0)\}$ is used. The weights that achieve this are obtained by iterative search:

$$\boldsymbol{w}_1(i+1) \leftarrow \boldsymbol{w}_1(i) - \frac{\mu}{2}\frac{\partial}{\partial \boldsymbol{w}_1(i)} E\{y_1^2(i)\} = \boldsymbol{w}_1(i) + \mu E\{y_1(i)\tilde{\boldsymbol{b}}_1(i)\}. \tag{7}$$

That is, by minimizing the output energy, the algorithm forces the correlation between the desired output signal and vector of tentative decisions of interfering signals to zero. The steady state values of the weights are found from $\frac{\partial}{\partial \boldsymbol{w}_1(0)} E\{y_1^2(0)\} = 0$ and are evaluated as:

$$\begin{aligned} -\frac{1}{2}\frac{\partial}{\partial \boldsymbol{w}_1(0)} E\{y_1^2(0)\} = \; 0 \; &= E\{y_1(0)\tilde{\boldsymbol{b}}_1(0)\} \\ &= E\{x_1(0)\tilde{\boldsymbol{b}}_1(0) - \boldsymbol{w}_1(0)\tilde{\boldsymbol{b}}_1(0)\tilde{\boldsymbol{b}}_1(0)^T\}, \end{aligned} \tag{8}$$

with

$$E\{x_1(0)\tilde{\boldsymbol{b}}_1(0)\} = E\{\sqrt{a_1}b_1(0)\tilde{\boldsymbol{b}}_1(0) + \boldsymbol{\rho}_1^T \boldsymbol{A}\boldsymbol{b}_1(0)\tilde{\boldsymbol{b}}_1(0) + n_1(0)\tilde{\boldsymbol{b}}_1(0)\}. \tag{9}$$

It is easy to show that the expectations of the first and third term in (9) are zero. We thus get:

$$E\{x_1(0)\tilde{\boldsymbol{b}}_1(0)\} = \boldsymbol{A}E\{\boldsymbol{b}_1(0)\tilde{\boldsymbol{b}}_1^T(0)\}\boldsymbol{\rho}_1. \tag{10}$$

Clearly $E\{\boldsymbol{b}_1(0)\tilde{\boldsymbol{b}}_1^T(0)\}$ is diagonal; therefore, the system of $(2K-1)$ linear equations (8), together with (10), gives the steady state values of the weights affecting the first output as:

$$\boldsymbol{w}_1(0) = \left[E\{\tilde{\boldsymbol{b}}_1(0)\tilde{\boldsymbol{b}}_1^T(0)\}\right]^{-1} \boldsymbol{A}E\{\boldsymbol{b}_1(0)\tilde{\boldsymbol{b}}_1^T(0)\}\boldsymbol{\rho}_1. \tag{11}$$

The expectations in the above expression are given in the Appendix.

2.1 Error Probability

The output probability of error P_{e_1} of user 1 is evaluated as:

$$\begin{aligned} P_{e_1} &= E_{b_1(0),\boldsymbol{b}_1(0),\tilde{\boldsymbol{b}}_1(0)} Pr\{\hat{b}_1 \neq b_1(0) \mid b_1(0), \boldsymbol{b}_1(0), \tilde{\boldsymbol{b}}_1(0)\} \\ &= \frac{1}{2}\sum_{\boldsymbol{b}_1(0)} E_{\tilde{\boldsymbol{b}}_1(0)}\left[Pr\{n_1(0) > \sqrt{a_1} - \boldsymbol{\rho}_1^T\boldsymbol{A}\boldsymbol{b}_1(0) + \boldsymbol{w}_1^T(0)\tilde{\boldsymbol{b}}_1(0) \mid \tilde{\boldsymbol{b}}_1(0)\}\right. \\ &+ \left. Pr\{n_1(0) < -\sqrt{a_1} - \boldsymbol{\rho}_1^T\boldsymbol{A}\boldsymbol{b}_1(0) + \boldsymbol{w}_1^T(0)\tilde{\boldsymbol{b}}_1(0) \mid \tilde{\boldsymbol{b}}_1(0)\}\right] Pr\{\boldsymbol{b}_1(0)\}. \end{aligned} \tag{12}$$

Since $Pr\{\boldsymbol{b}_1(0)\} = 2^{-(2K-2)}$, and the above expression contains pairwise identical terms, it can be written as:

$$P_{e_1} = \frac{1}{2^{2K-2}}\sum_{\boldsymbol{b}_1(0)} E_{\tilde{\boldsymbol{b}}_1(0)} Pr\{n_1(0) > \sqrt{a_1} - \boldsymbol{\rho}_1^T\boldsymbol{A}\boldsymbol{b}_1(0) + \boldsymbol{w}_1^T(0)\tilde{\boldsymbol{b}}_1(0) \mid \tilde{\boldsymbol{b}}_1(0)\}. \tag{13}$$

2.1.1 Dead-Zone Limiter

Focusing on the dead-zone soft tentative decision, the vector $\boldsymbol{\xi}_1$ spans a $(2K-2)$-dimensional space. Each dimension is partitioned as:

$$\begin{aligned} &-\lambda_{k1}^{(i)} - \sqrt{a_k}b_k(i) < \xi_k(i) < \lambda_{k1}^{(i)} - \sqrt{a_k}b_k(i) \\ &\text{or} \quad \xi_k(i) > \lambda_{k1}^{(i)} - \sqrt{a_k}b_k(i), \quad \xi_k(i) < -\lambda_{k1}^{(i)} - \sqrt{a_k}b_k(i), \end{aligned} \tag{14}$$

with $k = 2, 3, \ldots, K$, $i = -1, 0$. The created $(2K-2)$-dimensional subregions D_m, $m = 1, 2, \ldots, 2^{2K-2}$, have a corresponding vector $\tilde{\boldsymbol{b}}_1(0) = \tilde{\boldsymbol{b}}_{1m}(0)$ whose elements, according to (4), take values ± 1 and 0. Thus (13) can be rewritten as:

$$P_{e_1} = \frac{1}{2^{2K-2}}\sum_{\boldsymbol{b}_1(0)}\sum_{m=1}^{2^{2K-2}} Pr\{n_1(0) > \sqrt{a_1} - \boldsymbol{\rho}_1^T\boldsymbol{A}\boldsymbol{b}_1(0) + \boldsymbol{w}_1^T(0)\tilde{\boldsymbol{b}}_{1m}(0)\} Pr\{\boldsymbol{\xi}_1 \in \boldsymbol{D}_m\}. \tag{15}$$

Since $n_1(0)$ and $\boldsymbol{z}_1(0)$ are uncorrelated, we finally get:

$$P_{e_1} = \frac{1}{2^{2K-2}}\sum_{\boldsymbol{b}_1(0)}\sum_{m=1}^{2^{2K-2}} Q\left(\frac{\sqrt{a_1} - \boldsymbol{\rho}_1^T\boldsymbol{A}\boldsymbol{b}_1(0) + \boldsymbol{w}_1^T(0)\tilde{\boldsymbol{b}}_{1m}(0)}{\sqrt{N_0/2}}\right) \int_{\boldsymbol{D}_m} f_{\boldsymbol{\xi}_1}\, d\boldsymbol{\xi}_1, \tag{16}$$

where $f_{\boldsymbol{\xi}_1}$ is a $(2K-2)$-variate Gaussian density function.

2.1.2 Linear Clipper

In the case of a linear clipper, each entry of $\tilde{\boldsymbol{b}}_1(0)$ in (13) can, according to (5), be $z_k(i)/\lambda_{k1}^{(i)}$ or $sgn(z_k(i))$. A transformed Gaussian random variable ψ_1 is defined as:

$$\psi_1 = n_1(0) - \sum_{k=2}^{K} c_k^{(0)} \frac{w_{k1}^{(0)}}{\lambda_{k1}^{(0)}} \xi_k(0) - \sum_{k=2}^{K} c_k^{(-1)} \frac{w_{k1}^{(-1)}}{\lambda_{k1}^{(-1)}} \xi_k(-1),$$

where

$$c_k^{(i)} = \begin{cases} 1 & |z_k(0)| < \lambda_{k1}^{(i)} \\ 0 & \text{otherwise} \end{cases} \qquad k = 2, 3, \ldots, K, \quad i = -1, 0.$$

The error probability (13) becomes:

$$P_{e_1} = \frac{1}{2^{2K-2}} \sum_{\boldsymbol{b}_1(0)} E_{\tilde{\boldsymbol{b}}_1(0)} Pr\{\psi_1 > \sqrt{a_1} - \boldsymbol{\rho}_1^T \boldsymbol{A} \boldsymbol{b}_1(0) + \boldsymbol{w}_1^T(0) \boldsymbol{h}_1(0) \mid \tilde{\boldsymbol{b}}_1(0)\},$$

where $\boldsymbol{h}_1(0) = [h_2(-1), \ldots, h_K(-1), h_2(0), \ldots, h_K(0)]^T$ and each entry is defined as:

$$h_k(i) = \begin{cases} \sqrt{a_k} b_k(i)/\lambda_{k1}^{(i)} & |z_k(i)| < \lambda_{k1}^{(i)} \\ \text{sgn}(z_k(i)) & \text{otherwise} \end{cases} \qquad k = 2, 3, \ldots, K, \quad i = -1, 0.$$

Defining the vector $\boldsymbol{\zeta}$ as:

$$\boldsymbol{\zeta} = [\zeta_1, \zeta_2, \ldots, \zeta_{2K-1}]^T = [\psi_1, \xi_2(-1), \xi_2(0), \ldots, \ldots, \xi_K(-1), \xi_K(0)]^T,$$

that spans a $(2K-1)$-dimensional space, the final expression for the error probability becomes

$$P_{e1} = \frac{1}{2^{2K-2}} \sum_{\boldsymbol{b}_1(0)} \sum_{m=1}^{2^{2K-2}} \int_{\boldsymbol{D}_m} \boldsymbol{f}_{\boldsymbol{\zeta}} d\boldsymbol{\zeta}, \tag{17}$$

where $\boldsymbol{f}_{\boldsymbol{\zeta}}$ is a $(2K-1)$-variate Gaussian density function. Each dimension is partitioned as $\sqrt{a_1} - \boldsymbol{b}_1(0)^T \boldsymbol{A} \boldsymbol{\rho}_1 + \boldsymbol{w}_1^T(0) \boldsymbol{h}_1(0) < \zeta_1$, and for $j = 2, 3, \ldots, 2K-1$, each ζ_j falls in either of the following two regions:

$$\begin{aligned} &-\lambda_{k1}^{(i)} - \sqrt{a_k} b_k(i) < \zeta_j < \lambda_{k1}^{(i)} - \sqrt{a_k} b_k(i) \\ \text{or} \quad &\zeta_j > \lambda_{k1}^{(i)} - \sqrt{a_k} b_k(i), \zeta_j < \lambda_{k1}^{(i)} - \sqrt{a_k} b_k(i), \end{aligned} \tag{18}$$

again with $k = 2, 3, \ldots, K, \quad i = -1, 0$.

3 Numerical Examples and Discussion

In the examples the signal-to-noise ratio for user k is defined as $SNR_k = a_k/N_0$. For two asynchronous users, Figure 2 shows the probability of error of user 1 versus SNR_1. The relative energy e_1 is defined as $e_1 = \int_{\tau_2}^{T} s_1^2(t) dt$. The case corresponds to a rather weak level of interference which is a favorable scenario for this multistage detector due to its ability to reduce the interference causing unreliable tentative decisions. Thus the

performance of the detector is superior to the one utilizing hard tentative decisions. At higher SNR's the linear clipper slightly outperforms the dead zone limiter.

Because of the improved reliability of the tentative decisions, the hard limiter performs much better than in the previous case when the interference is relatively stronger, as shown in Figure 2. The linear clipper still slightly outperforms it, while the dead zone limiter is somewhat inferior. However, the variations in the error performance of the detector with hard tentative decisions are much larger in the above two scenarios than in the case with soft tentative decisions.

Figure 3 shows the probability of error of user 1 versus the relative interference energy for the fixed SNR_1. The improved performance of the detector that utilizes soft tentative decisions is demonstrated again.

Gold sequences of length 7, two of which are shown in Figure 4, are used as the signature sequences in the next example. Figure 5 shows the probability of error for user 1 when its energy and the energy of the interferer are the same. The worst case and the average error performance over the values of the relative delay τ_2 are shown. In this scenario the dead-zone limiter and the linear clipper perform almost indentically. The detector with the soft limiter clearly outperforms both the conventional detector and the detector with the hard tentative decision, and its average performance, due to the good cross-correlation properties of the two signature sequences used, is very close to the single-user bound.

4 Appendix

Let f_{z_k} be the density function of $z_k(i)$, and $\boldsymbol{f}_{z_k z_l}$ the joint density function of $z_k(i)$ and $z_l(i)$. For the entries in the matrix $E\{\tilde{\boldsymbol{b}}_1(0)\tilde{\boldsymbol{b}}_1^T(0)\}$ in (11), the diagonal elements in the case of a linear clipper are:

$$E\{\tilde{b}_k^2(i)\} = \frac{1}{2}\sum_{b_k(i)}\left[\int_{\boldsymbol{L}_1} f_{z_k}dz_k + \int_{\boldsymbol{L}_2}\frac{z_k^2(i)}{\lambda_{k1}^{(i)^2}}f_{z_k}dz_k\right],$$

while for the dead-zone limiter:

$$E\{\tilde{b}_k^2(i)\} = \frac{1}{2}\sum_{b_k(i)}\int_{\boldsymbol{L}_1} f_{z_k}dz_k.$$

The off-diagonal entries for the linear clipper are:

$$\begin{aligned} E\{\tilde{b}_k(i)\tilde{b}_l(i)\} &= \frac{1}{4}\sum_{b_k(i),b_l(i)}\left[\int\int_{\boldsymbol{Z}_1} sgn(z_k(i))sgn(z_l(i))\boldsymbol{f}_{z_k z_l}dz_k dz_l\right. \\ &+ \int\int_{\boldsymbol{Z}_2}\frac{z_k(i)}{\lambda_{k1}^{(i)}}sgn(z_l(i))\boldsymbol{f}_{z_k z_l}dz_k dz_l \\ &+ \left.\int\int_{\boldsymbol{Z}_3}\frac{z_l(i)}{\lambda_{l1}^{(i)}}sgn(z_k(i))\boldsymbol{f}_{z_k z_l}dz_k dz_l\right] \end{aligned}$$

$$+ \int\int_{\boldsymbol{Z}_4} \frac{z_k(i) z_l(i)}{\lambda_{k1}^{(i)} \lambda_{l1}^{(i)}} \boldsymbol{f}_{z_k z_l} dz_k dz_l \Bigg]$$

$$E\{b_k(i)\tilde{b}_k(i)\} = \frac{1}{2} \sum_{b_k(i)} \left[\int_{\boldsymbol{L}_1} b_k(i) sgn(z_k(i)) f_{z_k} dz_k + \int_{\boldsymbol{L}_2} b_k \frac{z_k(i)}{\lambda_{k1}^{(i)}} f_{z_k} dz_k \right],$$

and for the dead-zone:

$$E\{\tilde{b}_k(i)\tilde{b}_l(i)\} = \frac{1}{4} \sum_{b_k(i), b_l(i)} \int\int_{\boldsymbol{Z}_1} sgn(z_k(i)) sgn(z_l(i)) \boldsymbol{f}_{z_k z_l} dz_k dz_l$$

$$E\{b_k(i)\tilde{b}_k(i)\} = \frac{1}{2} \sum_{b_k(i)} \int_{\boldsymbol{L}_1} b_k(i) sgn(z_k(i)) f_{z_k} dz_k.$$

For the second expectation in (11), $E\{\boldsymbol{b}_1(0)\tilde{\boldsymbol{b}}_1^T(0)\}$, we have in the case of the linear clipper:

$$E\{b_k(i)\tilde{b}_k(i)\} = \frac{1}{2} \sum_{b_k(i)} \left[\int_{\boldsymbol{L}_1} b_k(i) sgn(z_k(i)) f_{z_k} dz_k + \int_{\boldsymbol{L}_2} b_k \frac{z_k(i)}{\lambda_{k1}^{(i)}} f_{z_k} dz_k \right],$$

and for the dead-zone:

$$E\{b_k(i)\tilde{b}_k(i)\} = \frac{1}{2} \sum_{b_k(i)} \int_{\boldsymbol{L}_1} b_k(i) sgn(z_k(i)) f_{z_k} dz_k,$$

where we assume $i, j = -1, 0, \quad k, l = 2, \ldots, K, \quad k \neq l$.
$\boldsymbol{L}_j$, $j = 1, 2$ corresponds to the appropriate intervals of z_k, and $\boldsymbol{Z}_j$, $j = 1, \ldots, 4$ corresponds to the appropriate rectangular regions in the (z_k, z_l) plane.

References

[1] M.K. Varanasi and B. Aazhang, "Multistage detector in asynchronous code-division multiple access communications," *IEEE Trans. Commun.*, vol. 38, No. 4, pp. 509-519, Apr. 1990.

[2] X. Zhang and D. Brady, "Soft-decision multistage detection for asynchronous AWGN channels," *Proc. 31st Annual Allerton Conf. Commun.*, Sept. 1993.

[3] Z. Siveski, L. Zhong and Y. Bar-Ness, "Adaptive multiuser CDMA detector for asynchronous AWGN channels," IEEE Int'l Symp. on Personal, Indoor and Mobile Radio Commun., The Hague, The Netherlands, Sept. 1994.

[4] S. Verdú, "Recent progress in multiuser detection," *Advances in Communication and Signal Processing*, Springer-Verlag, 1989.

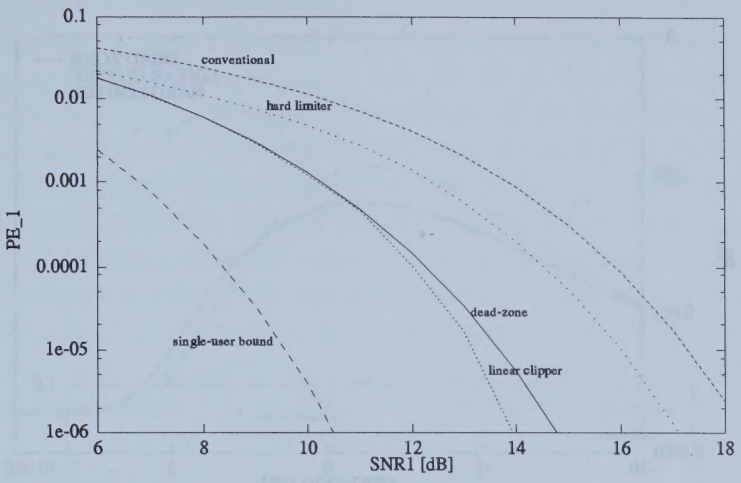


Figure 1: Error probability of user 1 for $K = 2$, $\rho_{12} = 0.2$, $\rho_{21} = 0.6$, $e_1 = 0.4$, $a_2/a_1 = 0.6$

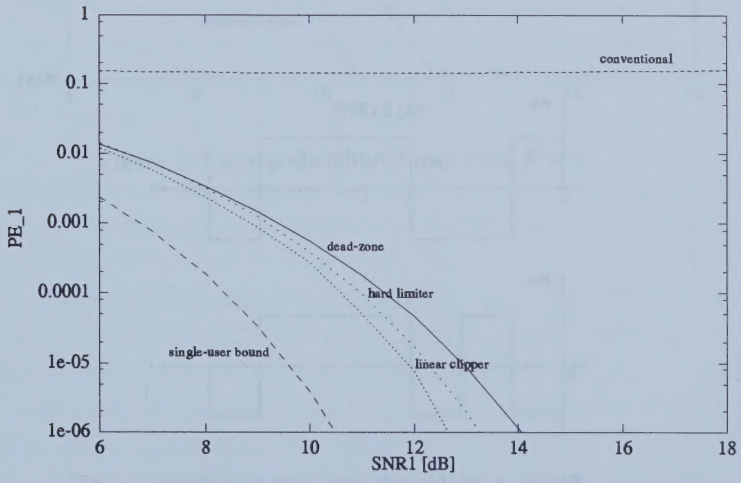


Figure 2: Error probability of user 1 for $K = 2$, $\rho_{12} = 0.2$, $\rho_{21} = 0.6$, $e_1 = 0.4$, $a_1/a_2 = 0.6$

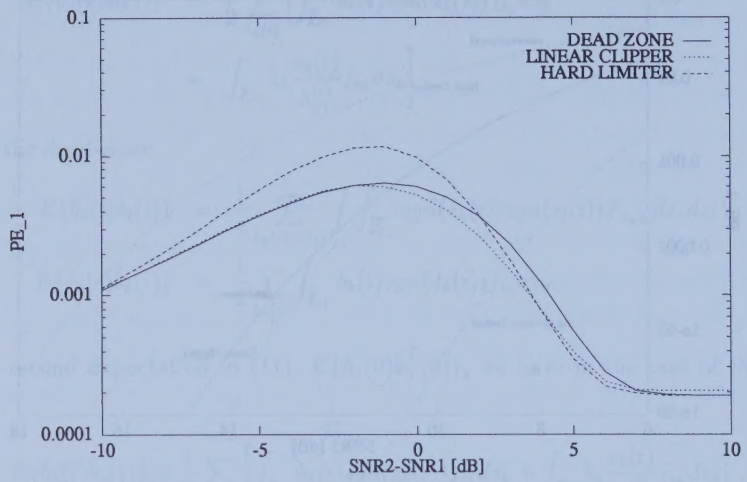


Figure 3: Error probability of user 1 for $K = 2$, $SNR_1 = 8\ dB$, $\rho_{12} = 0.2$, $\rho_{21} = 0.6$, $e_1 = 0.4$

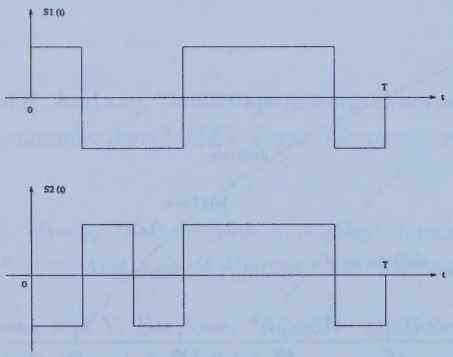


Figure 4: Gold-code signature waveforms

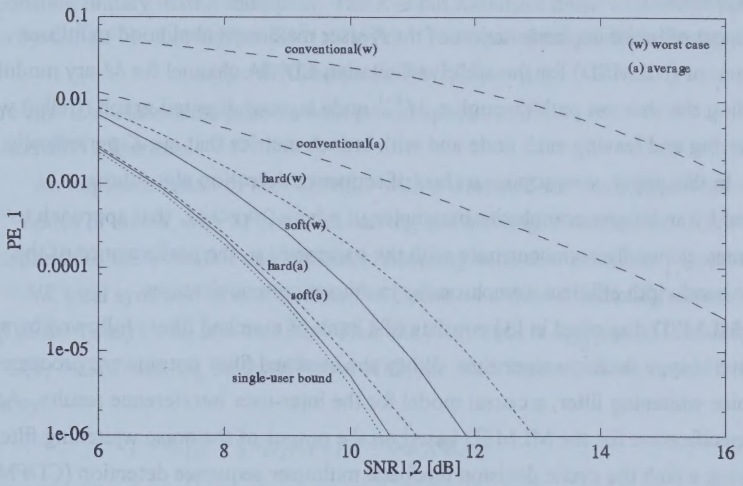


Figure 5: Error probability of user 1 for $K = 2$

Cyclic Decision Feedback Multiuser Sequence Detection

Eric A. Fain, STUDENT MEMBER, IEEE, and Mahesh K. Varanasi[1], MEMBER, IEEE

ABSTRACT-- The cyclic decision feedback multiuser sequence detector (CDFMSD) is proposed for the asynchronous Gaussian CDMA channel as a sub-optimum variant of the maximum likelihood detector (MLMSD). The CDFMSD employs a trellis detection algorithm with fewer states than the MLMSD. It is parameterized by ν, the number of feedback symbols, or past decisions stored at each state (or node) in the detection trellis. When $\nu=0$, the CDFMSD is identical to the MLMSD. The error performance and asymptotic efficiency of the detector are investigated, and numerical results are given.

INTRODUCTION

The most efficient implementation of the K-user maximum likelihood multiuser sequence detector (MLMSD) for the additive Gaussian CDMA channel for M-ary modulation involves finding the shortest path through a M^{K-1} node layered directed graph (trellis) with M branches entering and leaving each node and with branch metrics that are K-periodically time-varying [3]. In this paper, we propose a class of sequence detection algorithms, parameterized by an integer complexity parameter μ, where $0<\mu<K-1$, that approach to varying degrees, generally commensurate with the parameter μ, the performance of the MLMSD for bandwidth efficient asynchronous multiuser communications.

The MLMSD described in [3] consists of a bank of matched filters followed by a multiuser Viterbi-type decision algorithm. When the matched filter outputs are processed by a multiuser noise whitening filter, a causal model for the inter-user interference results. An equivalent specification for the MLMSD based on the output of the noise whitening filter is given following which the cyclic decision feedback multiuser sequence detection (CDFMSD) algorithm is described, which is inspired by the delayed decision feedback sequence estimator (DDFSE) proposed in [1] for the single-user ISI channel. The error performance of the CDFMSD is estimated, and corroborating numerical results are given.

SYSTEM MODEL

The fundamental enabling result here is the multivariate spectral factorization of a spectral polynomial (more generally, rational) transfer matrix $\mathbf{S}(z)$ obtained by Wiener and Masani in [5] where the authors establish the existence and uniqueness (modulo multiplication by a unitary matrix), of a minimum-phase polynomial (rational) transfer matrix $\mathbf{G}(z)$ so that $\mathbf{S}(z) = \mathbf{G}^{\mathrm{T}}(z^{-1})\mathbf{G}(z)$. A Newton-Raphson method for this computation of $\mathbf{G}(z)$ is given in [6]. Denote the output sequence of the K users' matched filters as $\{\mathbf{y}(i)\}$ where $y_k(i)$ is the i^{th} output of the k^{th} user's matched filter. Let $\{\mathbf{b}(i)\}$ be the vector sequence of symbols

[1]The authors are with the Department of Electrical and Computer Engineering at the University of Colorado, Boulder, CO 80309. This work was supported by NSF grant NCR-9206327

transmitted by the K users with $b_k(i)$ being the i^{th} symbol of the k^{th} user. For the additive white Gaussian channel, the discrete-time signal-plus-noise model is given as $\mathbf{y}(i)=(\mathbf{H} * \mathbf{b})(i) + \mathbf{n}(i)$, where $*$ denotes the convolution operator, and $\{\mathbf{H}(i)\}$ is the impulse response of the equivalent K-input K-output linear time-invariant multiuser transmitter filter. Assuming an ideal channel with no ISI, and numbering the users in order of their increasing relative time delays, $\mathbf{H}(i) = \mathbf{0}$, $|i| > 1$ (i.e. the transmitter filter is FIR, and furthermore, $\mathbf{H}(1)$ is strictly upper triangular with $\mathbf{H}(-1)=\mathbf{H}^{T}(1)$, and $\mathbf{H}(0)$ symmetric). Letting the matrix z-transform of the impulse response be $\mathbf{S}(z)=\mathbf{H}(-1)z + \mathbf{H}(0) + \mathbf{H}(1)z^{-1}$, the power spectrum of the vector additive noise sequence $\{\mathbf{n}(i)\}$ is $\sigma^2\mathbf{S}(z)$. Using the multivariate spectral factorization result and exploiting the special structure of our matrix $\mathbf{S}(z)$, we can obtain a minimum phase factor $\mathbf{G}(z)$ that is first-order FIR ($\mathbf{G}(z)$ has the form $\mathbf{G}(0)+\mathbf{G}(1)z^{-1}$), and moreover, $\mathbf{G}(0)$ can be chosen to be lower triangular and $\mathbf{G}(1)$ strictly upper triangular by an appropriate choice of the constant unitary matrix multiplier. The K-input K-output noise whitening filter is then given by $\mathbf{G}^{-T}(z^{-1})$. Let $\mathbf{w}(z) = \mathbf{G}^{-T}(z^{-1})\mathbf{y}(z)$. The equivalent model is then given by:

$$\mathbf{w}(i) = \mathbf{G}(0)\mathbf{b}(i) + \mathbf{G}(1)\mathbf{b}(i-1) + \eta(i) \tag{1}$$

where $\eta(i)$ is a white noise process with power spectrum $\sigma^2\mathbf{I}$.

The MLMSD Algorithm

An implementation of the MLMSD with a whitened matched filter front-end will have M^{K-1} nodes or states, with M branches leaving and arriving at each node. Each node corresponds to the arrival of one symbol. Assuming each user transmits N symbols, there will be $J = NK$ total symbols. It is convenient to define a modulo-K indexing scheme where $j=\kappa(j) + \psi(j)K$ for $1\le \kappa(j) \le K$, and $1\le j \le J$. Thus the j^{th} symbol is $b_{\kappa(j)}(\psi(j))$. Additionally, letting $\mathbf{G} = \mathbf{G}(0) + \mathbf{G}(1)$ and $\mathbf{G}_k$ be the k^{th} row of $\mathbf{G}$, the branch metric from state j-1 to state j is given by:

$$c_j = c_{\kappa(j)}^{\psi(j)} = \left(w_{\kappa(j)}(\psi(j)) - \mathbf{G}_{\kappa(j)}\mathbf{q}(j)\right)^2 \tag{2}$$

where $\mathbf{x}(0) = \mathbf{0}$, $\mathbf{x}(j) = \left[x_1(j-1)\cdots x_{\kappa(j)-1}(j-1)\ \hat{b}_{\kappa(j)}(\psi(j))\ x_{\kappa(j)+1}(j-1)\cdots x_K(j-1)\right]^T$, and $\hat{b}$ is a candidate for b. The path metric is defined as the sum of the corresponding branch metrics for a node sequence. The survivor path is retained at each node and is defined as the one having the smallest path metric to that node. The maximum likelihood path will be the survivor path at the termination node. Typically, however, the sequence decisions can be made after a sufficient fixed delay [2],[3]. The MLMSD trellis detection algorithm is depicted in Figure 1 for a three-user channel with binary modulation.

The CDFMSD Algorithm

The CDFMSD, shown in Figure 2, utilizes the same matched filter bank and whitening filter front-end as the MLMSD. The reduction in complexity comes in the trellis detection (CDFTD) algorithm . This trellis consists of M^{μ} states where $\mu < K$-1 is the complexity constraint. Whereas each node in the MLMSD trellis represents the past K-1 symbols leading to that node, the CDFTD nodes represent the past μ symbols, and also store the remaining ν =

K-1 - μ symbols of the survivor path leading to that node. The branch metrics from state j-1 to state j are given by:

$$\lambda_j = \lambda_{\kappa(j)}^{\psi(j)} = \left(w_{\kappa(j)}(\psi(j)) - \mathbf{G}_{\kappa(j)}\mathbf{q}(j)\right)^2 \quad (3)$$

The path history $\mathbf{q}(j)$, is updated in a cyclic manner according to:

$$\mathbf{q}(j+1) = \left[q_1(\mathrm{j}) \cdots q_{\kappa(j)}(\mathrm{j})\ \hat{b}_{\kappa(j+1)}(\psi(j))\ a_1(\mathrm{j}) \cdots a_\nu(\mathrm{j})\ q_{\kappa(j+1)+\nu+1}(\mathrm{j}) \cdots q_K(\mathrm{j})\right]^T \quad (4)$$

when $\nu \leq K$ - κ(j+1), and when $\nu > K$ - κ(j+1),

$$\mathbf{q}(j+1) = \left[a_{K-\kappa(j+1)+1}(\mathrm{j}) \cdots a_\nu(\mathrm{j})\ q_{\nu-K+\kappa(j+1)+1}(\mathrm{j}) \cdots q_{\kappa(j)}(\mathrm{j})\ \hat{b}_{\kappa(j+1)}(\psi(\mathrm{j}))\ a_1(\mathrm{j}) \cdots a_{K-\kappa(j+1)}(\mathrm{j})\right]^T$$

where $\mathbf{q}(0)=\mathbf{0}$, $\mathbf{a}(0) = \mathbf{0}$, and $\mathbf{a}(j)$ denotes the ν decision-feedback symbols stored with each node, or the path history to that node from time j-K+1 to j-K+ν.

Once the survivor paths to each of the current nodes are obtained using the branch metrics just described, the feedback decisions stored in each current node are those that are associated with the survivor path up to that node. These past decisions are kept for each survivor path arriving at each node, and are not all the same, thus they do not reflect hard decisions fed back from the detector output. The number of nodes in each layer of the trellis is equal to $2^{K-\nu-1} = 2^{\mu}$. The CDFTD algorithm is depicted in Figure 2 for the three-user binary channel, with $\mu = 1$, so that the four-state MLMSD algorithm of Figure 1 is now replaced with a two-state trellis.

PERFORMANCE ANALYSIS

The MLMSD has error performance well characterized by the k^{th} user's minimum distance error event, and a tight error bound is given in [3]. The CDFMSD is not amenable to such an exact analysis. This is due to the fact that the set of survivor paths kept at each layer transition is a randomly selected subset of all possible survivor paths. Under certain assumptions, the performance can be estimated by considering two classes of errors: primary events, which are independent of each other, and secondary events, which are dependent upon a previous primary event.

A. Estimating probabilities from primary error events:

The *error alphabet* is the set of symbols which can represent the difference between the transmitted and received symbols. We shall consider binary antipodal modulation, so that only {1,-1} are transmitted, thus the error alphabet is {2,0,-2}. The *error trellis* will have a structure similar to the detection trellis, except that since there are three error symbols, there will be 3^{μ} nodes in each layer. Each layer transition corresponds to the arrival of one user's bit at the detector. The *ground* node is the all-zero state which corresponds to the past μ symbols being correctly detected. An *error event* is defined as a path through the error trellis which starts and ends in the ground node. This corresponds to an excursion from the correct node path in the detection trellis.

The error symbol sequence associated with an error event, denoted as $\mathbf{e}$, is drawn from the set {-2,0,2} and has the structure $\mathbf{e}$=[$\mathbf{e}_0$ 0,...,0], where $\mathbf{e}_0$ is followed by μ zeros, and $\mathbf{e}_0$

begins and ends in a non-zero value and contains no more than μ-1 consecutive zeros. A primary error event is one which is preceded by at least K-1 zeros. All error events in the MLMSD are primary, since every error event ends in K-1 zeros.

Let **e** be the sequence associated with a primary error event. The probability of an error in the k^{th} user's symbol occurring from primary error event **e** can be estimated by considering the conditional probability P(**e**) that **b**-**e** is chosen as the correct path, conditioned on **b** being a candidate path at the final node in the erroneous sequence. Then P(**e**) = P(**e** is allowable & the error path survives, given that the correct path is a candidate path at the final node of the error event). The sequence **e** is allowable if the transmitted symbols are commensurate with the error symbols $\{e_j\}$. The probability that **e** is allowable is $2^{-w(\mathbf{e})}$ for the binary modulation case, where w(**e**) is the Hamming weight of **e**. The sequence **b**-**e**, **b** being the correct symbol sequence, will be chosen over **b** if the noise components exceed half of the distance between **e** and the all-zero path. In the presence of Gaussian noise with variance σ^2, this will occur with probability $Q(d_\mu(\mathbf{e})/2\sigma)$. This distance is defined as:

$$d_\mu^2(\mathbf{e}) = \sum_{j=j_0}^{j_0+l-1} \sum_{k=0}^{K-1} (\mathbf{G}_{\kappa(j),\kappa(j-k)} \cdot e_{j-k})^2 \qquad (5)$$

Where $\mathbf{G}_{mn}$ is the $(m,n)^{th}$ element of $\mathbf{G}_0 + \mathbf{G}_1$, and sequence $\mathbf{e}_0$ starts at time j_0 and has length l. This squared distance of the error event is the error trellis path metric for the error sequence. The squared distance would also be the energy received if the error sequence itself was transmitted. The estimate of the k^{th} user error probability based on the consideration of primary events alone may be given as:

$$P_k^{(p)} = \sum_{\mathbf{e}\in\mathbf{A}} N_\mathbf{e} \cdot 2^{-w(\mathbf{e})} \cdot Q(\tfrac{d_\mu(\mathbf{e})}{2\sigma}) \qquad (6)$$

where $N_\mathbf{e}$ is the number of errors affecting the k^{th} user in the event **e**, and **A** is the set of all primary error events affecting the k^{th} user. The error probability in (6) is the simple sequence upper bound for the maximum-likelihood case. Note that considering only primary events is not the same as assuming that all feedback decisions are correct.

The summation given above will be dominated by the term with the minimum d(**e**) for small σ. Thus, under the assumption of only primary error events occurring, as $\sigma \to 0$, the k^{th} user's asymptotic efficiency η_k can be can be estimated from (6) to be $(d_\mu(\mathbf{e})_{min})^2/w_k$, where w_k is the energy of the k^{th} user. When μ=K-1, this is equal to the asymptotic efficiency of the MLMSD [3].

In order to find the minimum-distance primary error event for the k^{th} user of the MLMSD, one must search over all valid paths through the error trellis. We can use an algorithm which searches K error trellises in parallel, each starting in a different layer. When the path metric at any node is greater than the maximum of the K candidate paths' minimum distances, that node may be deactivated for the next iteration. Branch metrics need only be calculated from those nodes which are active. The algorithm ends when all nodes are inactive,

or after $K(3^{K-1})$ iterations. To take advantage of the symmetry of the error events, all but one of the nodes in each initial layer can be deactivated.

For the CDFMSD, the algorithm must also search over the MLMSD error trellis (3^{K-1} nodes) because searching over the CDFMSD trellis may cause valid paths to be discarded. The candidate error paths must end at nodes denoted as *pseudo-ground* nodes, nodes to which exactly μ of the most recent consecutive symbols have been zero. At each layer transition (iteration), there will be $2K(3^{\nu})$ survivor paths arriving at the pseudo-ground nodes, each of which is a candidate for the minimum distance path affecting one or more users.

At moderate SNR, it may also be necessary to find the second smallest error event for each user. To search for both the first and second smallest events, it is necessary to save the two smallest survivor paths at each node (including the pseudo-ground nodes). Nodes are then deactivated when the smallest survivor path exceeds the maximum of the second-smallest candidate paths.

B. Estimating error probabilities by including compound error events

Let $\mathbf{e}$ be an error sequence corresponding to a primary event, and $\mathbf{f}$ be a sequence corresponding to a secondary event. It is convenient to define the compound error sequence $\mathbf{ef}$ as the composition of four sub-sequences such that: $[\mathbf{e}\,\mathbf{f}] = [\mathbf{e}_1\ \mathbf{e}_2\ \underbrace{0\cdots0}_{\mu}\ \mathbf{e}_3\ \mathbf{e}_4]$ where $[\mathbf{e}_1\ \mathbf{e}_2]$ has less than μ consecutive zeros, and $\mathbf{f} = [\mathbf{e}_3\ \mathbf{e}_4]$ has less than μ consecutive zeros before its last non-zero element and ends in μ zeros. Also, $\mathbf{e}_2$ and $\mathbf{e}_3$ have length ν, and $\mathbf{e}_2$ ends in a non-zero value. If $\mathbf{e}_1$ has length greater than zero, then its first element will be non-zero. Lastly, $\mathbf{e}_4$ can be all-zeros or have length zero. Consider error event $\mathbf{f}$ of length n which begins at time $j_1 = j_0 + m$ following a primary event $\mathbf{e}$ of length l which begins at time j_0 $(0 \le m - l \le \nu - 1)$. Let $P(\mathbf{ef}) = P(\mathbf{f}|\mathbf{e})P(\mathbf{e})$ be the joint probability of $\mathbf{e}$ & $\mathbf{f}$, conditioned on the correct path being a candidate at time j_0+l, and on the correct path from time j_1 to $j_1 + n - 1$ being a candidate path at time $j_1 + n$. $P(\mathbf{ef})$ can be found by computing the residual interference of primary sequence $\mathbf{e}$ and its contribution to the distance of $\mathbf{f}$. The residual interference originates from feedback decisions stored in the ground node after the completion of a primary error event. These can be interpreted as an error vector which affects both the subsequent correct path and the error path equally. This residual sequence ρ can now be defined as

$$\rho_j = \sum_{k=0}^{K-1} \mathbf{G}_{\kappa(j),\kappa(j-k)} \cdot (\mathbf{e}_1)_{j-k} \qquad j = j_1,\ldots,j_1+\nu-1 \tag{7}$$

and let $|\mathbf{d}|$ be $\sqrt{\mathbf{d}^{\mathrm{T}}\mathbf{d}}$, the norm of $\{d_j\}$, where

$$d_j = \sum_{k=0}^{K-1} \mathbf{G}_{\kappa(j),\kappa(j-k)} \cdot b_{j-k} \qquad j = j_1,\ldots,j_1+n \tag{8}$$

In a primary error event, ($\rho = \mathbf{0}$), the distance from the correct path trajectory to the decision boundary is $|\mathbf{d}|/2$. For $|\rho| > \mathbf{0}$, the decision boundary is translated by the length of the projection of ρ onto $\mathbf{d}$, which is $\mathbf{d}(\rho^{\mathrm{T}}\mathbf{d}/|\mathbf{d}|)$. The equivalent distance of the secondary error

event is then $d_f = |\mathbf{d}| + 2\frac{\rho^T \mathbf{d}}{|\mathbf{d}|}$. When d_f is positive, $P(\mathbf{f}|\mathbf{e}) = 2^{-w(\mathbf{f})} Q(d_f/2\sigma)$. However, if d_f is negative, then the error path will be more likely than the correct path, and the Q-function will approach one very quickly, making $P(\mathbf{f}|\mathbf{e}) \approx 2^{-w(\mathbf{f})}$. Thus, if **e** is the minimum distance error sequence for the user(s) affected by **f**, then $P(\mathbf{e},\mathbf{f})$ will be on the order of $P(\mathbf{e})$, and its contribution will never vanish relative to $P(\mathbf{e})$. Certainly, this analysis could be extended to include those events which are themselves dependent upon secondary error events, but we will assume for now that such tertiary events will occur with low probability. (But this is not true for the case where $\mathbf{e} = \mathbf{f}$ is the minimum distance error event.) Under this assumption, the error probability for the k^{th} user of the CDFMSD is estimated as:

$$P_k^{(c)} = \sum_{\mathbf{e}\in\mathbf{A}} P(\mathbf{e}) N_{\mathbf{e}} + \sum_{\substack{\mathbf{f}\in\mathbf{B} \\ \mathbf{f}\in\mathbf{A}}} P(\mathbf{ef})(N_{\mathbf{e}} + N_{\mathbf{f}}).$$

Even if the largest contribution from the summation of compound events is on the order of the minimum distance event, the error probability at high SNR will still approach $cP(\mathbf{e}_{min})$ for some constant c, where $\mathbf{e}_{min} = \arg\min d_\mu^2(\mathbf{e})$, thus it will be reasonable to assume that the estimate of the asymptotic efficiency remains unaltered.

NUMERICAL RESULTS AND CONCLUSIONS

The numerical examples considered in this section are for a four-user channel, with users 1-4 modulating the length-seven signature sequences {+--+++-}, {++-----}, {-+-+++-}, and {+----+-}, respectively. Let τ ={delay 2, delay 3, delay 4} be the relative delays of users 2, 3, and 4 in chip intervals, and let w ={w_1,w_2,w_3,w_4} be the energies of the users. The asymptotic efficiency of user i, as defined above, is plotted in figures 3, 4, and 5 versus the ratio in dB of the energy of each interfering user to the energy of user i. All other users are assumed to have equal energies. The parameter ν is the number of feedback symbols per node, thus ν=0 corresponds to the MLMSD. Figure 3 shows the asymptotic efficiency of user 1 for τ ={1,4,5}. The ordering of the performance is commensurate with the complexity parameter μ. This performance is either superior or inferior to the decorrelator, depending upon the operating region. There are, of course, an infinite number of energy realizations that can be considered for this analysis. Since the asymptotic efficiency of the decorrelator is equal to the worst case of the MLMSD, there will always be a operating region for which the decorrelator will out-perform the CDFMSD. However, there will also be regions for which the CDFMSD will be very close to the MLMSD, and quite superior to the decorrelator. Figure 4 is shown for user 3, τ ={2,4,6}. In this particular case, the CDFMSD with ν =1 achieves unit efficiency.

Figure 5 is given for user 1, τ ={2,4,6}. Note that in this case, the ordering of performance is not always preserved. The explanation for this phenomenon is that the minimum distance error event in the CDFMSD may not contain more than μ-1 consecutive zero symbols. Thus, for small μ, the minimum-distance error sequence for a more complex

detection trellis may be inadmissible. Infrequently, the smallest admissible sequence may have a larger distance than the minimum-distance sequence for larger μ.

Error Probabilities:

Simulated and predicted curves for the probability of symbol error are given in figures 6-8. The predicted curves are generated from the first three terms in the summation given in (5). These first three terms are necessary for a close approximation to the error rate at the low to medium signal-to-noise ratios shown. Figures 6 represents the error rate of user 1, $\tau = \{1,4,5\}$, $w = \{1,1,1,1\}$. Figure 7 shows user 1, $\tau = \{1,4,5\}$, $w = \{1, .64,1,1\}$. Figure 8 depicts the simulated performance of user 2 for $\tau = \{1,4,5\}$ when $w = \{1, 1, .5, .5\}$. In this case, the feedback symbols for user 2's bits are at low reliability. This potentially undesirable condition does not cause a catastrophic increase in the error rate of the CDFMSD relative to the MLMSD since the feedback symbols do not reflect hard decisions about the path history.

References

[1] A. Duel-Hallen and C. Heegard, "Delayed decision-feedback sequence estimation," *IEEE Trans. Communications*, Vol. 37, No. 5, pp. 428-436, May, 1989.

[2] G.D. Forney, Jr., "Maximum-likelihood sequence estimation of digital signals in the presence of intersymbol interference," *IEEE Trans. Information Theory*, Vol. IT-18, No. 3, pp. 363-378, May, 1972.

[3] S. Verdu, "Minimum probability of error for asynchronous Gaussian multiple-access channels," *IEEE Trans. Information Theory*, Vol. IT-32, No. 1, pp. 85-96, Jan. 1986.

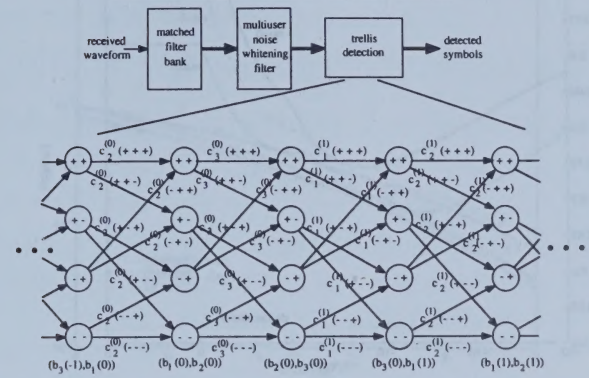
[4] _____, "Optimum multiuser asymptotic efficiency," *IEEE Trans. Communications*, Vol. COM-34, No. 9, pp. 890-897, Sept. 1986.

[5] N. Wiener, and P. Masani, "The prediction theory of multivariate stochastic processes, Part I," *Acta Math.*, 98, Nov. 1957, pp. 111-150.

[6] G.T.Wilson, "The factorization of matricial spectral densities," *SIAM J. Appl. Math.* Vol. 23, No. 4, 1972, pp. 420-426.

FIGURE 1

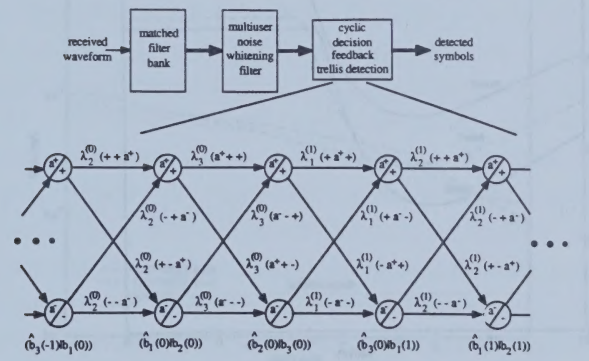
Maximum Likelihood Multiuser Sequence Detection



Trellis Detection for Three-User Gaussian CDMA Communication

FIGURE 2

Cyclic Decision Feedback Multiuser Sequence Detection



Cyclic Decision Feedback Trellis Detection (CDFTD)

FIGURE 3

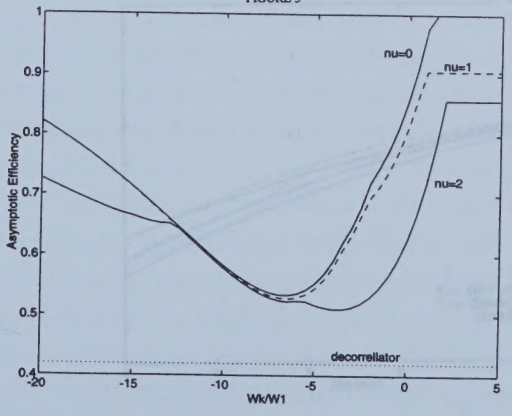


FIGURE 4

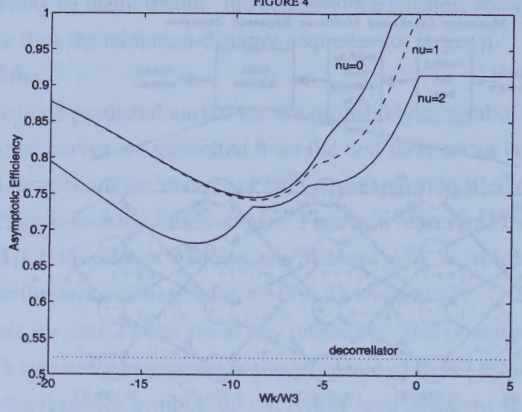


FIGURE 5

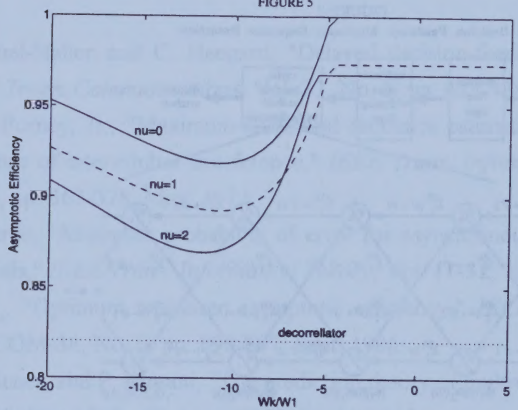


FIGURE 6

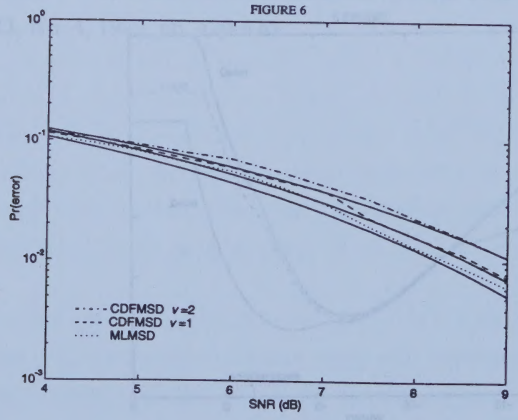


FIGURE 7

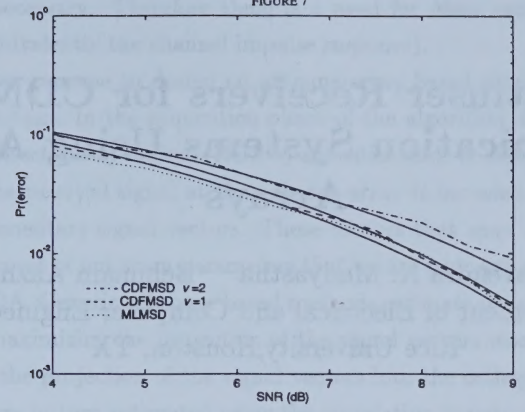
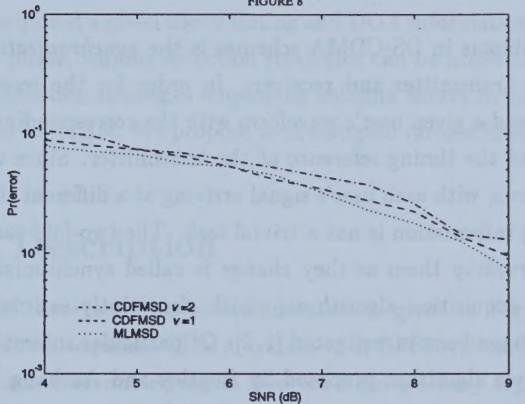


FIGURE 8



Multiuser Receivers for CDMA Communication Systems Using Antenna Arrays

Raghavendra K. Madyastha Behnaam Aazhang
Department of Electrical and Computer Engineering
Rice University,Houston, TX

1 Introduction

One of the main problems in DS-CDMA schemes is the *synchronization* of the code sequences used at the transmitter and receiver. In order for the receiver to accurately demodulate or *despread* a given user's waveform with the corresponding code, it needs an accurate estimation of the timing reference at the transmitter. Since the mode of transmission is asynchronous, with each user's signal arriving at a different time at the receiver, acquiring this timing information is not a trivial task. The two-fold task of acquiring the users' timings and *tracking* them as they change is called synchronization. Many computationally intense acquisition algorithms, which also jointly estimate amplitudes and phases for all users, have been investigated [1, 2]. Of particular interest to this study is a computationally simple algorithm proposed by Bensley and Aazhang [3–5]. We propose to extend this algorithm to enable acquisition, and thus detection, of more users than currently feasible with little loss in performance. We propose to exploit *spatial diversity* through the physical separation of more than one sensor or antenna used to receive the incoming signal. One immediate issue of spatial diversity is the design of *directional* antennas that can point their beams in a specific direction and transmit or receive information in a neighborhood of that direction only. This can result in a significant reduction in the interference level for a given user.

Antenna-array based acquisition of spread spectrum signals has been addressed before in [6, 7]. These studies deal with the maximum likelihood acquisition of a single user's timing in the presence of interference which is modeled as a stationary and spatially colored Gaussian process, uncorrelated in time. While single-user schemes are implementable in certain situations of low traffic, where users' timings are acquired sequentially, in general we require the simultaneous acquisition of multiple users to improve system performance. In addition, most single-user techniques assume the transmission of a *training sequence* to acquire the user's timing, which limits the data rate of the system, particularly if frequent

re-acquisition is necessary. Therefore there is a need for *blind* estimation of the signal parameters (or equivalently the channel impulse response).

In this study we propose to design an antenna-array based receiver for multiuser acquisition and detection. In the acquisition phase of the algorithm different users' timing information and *directions of arrival* (DOA's) are estimated. A subspace based approach is used wherein the received signal at the antenna-array is modeled as a linear combination of certain elementary signal vectors. These vectors that span a *signal subspace* are parameterized by a set of unknown parameters that we are trying to estimate, in this case, the timing and DOA. Generic subspace based methods estimate these parameters in either of two ways: by maximizing the projection of the signal vectors into the signal subspace, or by minimizing the projection of the signal vectors into the orthogonal *noise* subspace. These subspaces are in turn estimated using the correlation matrix of the received signal. In this paper we use the MUSIC algorithm [8,10–13] to estmate the timing and DOA.

Once we have acquired a given user's timing and DOA information, as well as estimated his amplitude and phase, various detection strategies can be implemented to estimate the user's data bits. Detection strategies employing antenna arrays in mobile cellular systems have been discussed in [14,15]. We propose to investigate various linear multiuser detectors in this regard.

2 System Description

A block diagram of the system under consideration is given in Fig 1. We assume a K-user narrow band direct sequence CDMA system with BPSK (Binary Phase Shift Keying) modulation with each transmitted signal selected from a binary alphabet and limited to $[0, T]$, where T is the symbol period. The narrow band assumption implies that the carrier frequency is much larger than the message bandwidth. We also make the assumption that each user's bit sequence is a zero mean stationary random vector with independent and identically distributed (i.i.d) components and that different users are independent of each other. Each user in the system "sees" a different channel in transmitting to the receiver. In this paper we assume a single path channel for a given user (extension to multipath channels is also possible) of the form

$$h_k(t) = \alpha_k \, \delta(t - \tau_k) \, , \tag{1}$$

where $\alpha_k \in \mathcal{C}$ is the complex channel attenuation and $\tau_k \in [0, T]$ is the corresponding path delay. Here we assume that the channel parameters do not vary in the time taken to estimate them.

The receiver is a uniform linear antenna array (ULA) (see Fig 2) with M sensors in a straight line, uniformly separated by a distance d. The total equivalent baseband signal received at the l^{th} sensor due to the superposition of all the signals from the K users,

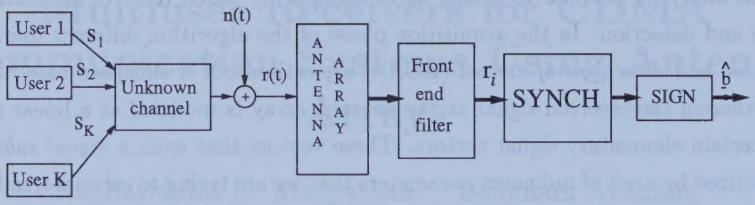


Figure 1: Block Diagram of the System

$r^{(l)}(t)$ is

$$r^{(l)}(t) = \sum_{k=1}^{K} r_k^{(l)}(t) \quad + \quad N_t^{(l)}, \quad t \in \mathbb{R}. \tag{2}$$

Here $N_t^{(l)}$ is the additive Gaussian noise at each sensor, assumed stationary and uncorre-

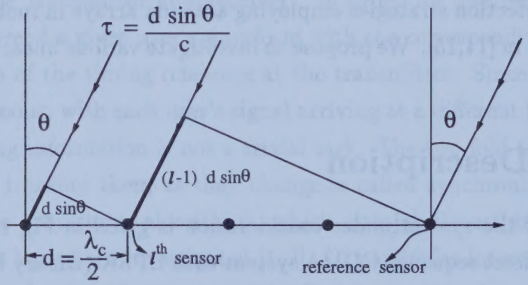


Figure 2: Schematic of a Uniform Linear Array with 5 elements

lated in time and space. Each user component $r_k^{(l)}(t)$ can be modeled as the convolution of the transmitted signal with a channel impulse response as

$$r_k^{(l)}(t) = \left(s_k(t) \otimes h_k^{(l)}(t)\right) e^{j\omega_c \Delta\tau_k^{(l)}}, \tag{3}$$

where the complex baseband representation of the k^{th} user's signal with BPSK modulation for the data, $s_k(t)$, is

$$s_k(t) = \beta_k \sum_i b_{k,i}\, a_k(t - iT), \tag{4}$$

with the k^{th} user's i^{th} bit, $b_{k,i} \in \{-1, 1\}$ and the complex number β_k representing the contribution due to the transmitted power and carrier phase relative to the oscillator at the receiver. The spreading waveform $a_k(t) = \sum_{j=0}^{N_c-1} a_{k,j} \Pi_{T_c}(t - jT_c)$, with $\Pi_{T_c}(t) = 1$, $t \in [0, T_c)$ and the chip or spreading sequence $a_{k,j} \in \{-1, 1\}$. The time period T_c is a chip duration with $T = N_c T_c$.

The term $e^{j\omega_c \Delta\tau_k^{(l)}}$ is a relative phase shift that accounts for the difference in the times of arrival, $\Delta\tau_k^{(l)}$, of the k^{th} signal at the reference and the l^{th} sensors. This relatively simple relationship among the received signals at different sensors is a result of the narrowband assumption of the signals. For this reason also, the impulse response for user k to sensor l, $h_k^{(l)}(t)$ is assumed to be independent of sensor, i.e., $h_k^{(l)}(t) = h_k(t)\ \forall l$. As seen in Fig 2, in the case of a ULA the differences in arrival times between adjacent sensors, $\Delta\tau_k = d\sin(\theta_k)/c$ where c is the speed of light and therefore $\Delta\tau_k^{(l)} = (l-1)d\sin(\theta_k)/c$. We define the "electronic" angle, ϕ_k, associated with the k^{th} signal as $\phi_k = \omega_c d\sin(\theta_k)/c$. If d is half a carrier wavelength, which is often the case to avoid spatial aliasing, $\phi_k = \pi\sin(\theta_k)$ and $\Delta\tau_k^{(l)} = (l-1)\phi_k$.

The received continuous time signal at each sensor is discretized by sampling the output of a filter matched to the chip waveform at the chip rate. The chip-matched filter in the case of BPSK data modulation amounts to a simple integrate and dump operation. Thus we get a discrete sequence at the l^{th} sensor given by

$$r^{(l)}(n) = \frac{1}{T_c}\int_{nT_c}^{(n+1)T_c} r^{(l)}(t)\,dt, \tag{5}$$

where the starting time of integration is chosen arbitrarily since the timings of the users in unknown. An observation vector, $\underline{r}_i^{(l)} \in \mathbb{C}^{N_c \times 1}$, is then generated at the l^{th} sensor after i bit intervals as $\underline{r}_i^{(l)} = [r^{(l)}(iN_c), r^{(l)}(iN_c+1), \ldots, r^{(l)}(iN_c+N_c-1)]^\top$. Because of the misalignment in integration arising out of the arbitrary reference time chosen, each N_c-vector $\underline{r}_i^{(l)}$ will contain a part of the current bit (the right part) and a part of the past bit (the left part) of each user [3]. We can then write $\underline{r}_i^{(l)}$ as

$$\underline{r}_i^{(l)} = \sum_{k=1}^{K}\left(\tilde{\beta}_{k,i-1}\,\underline{u}_k^L + \tilde{\beta}_{k,i}\,\underline{u}_k^R\right)e^{j(l-1)\phi_k} + \underline{N}_i^{(l)} = \sum_{k=1}^{2K} d_{k,i}\,\tilde{\underline{u}}_k\,e^{j(l-1)\phi_k} + \underline{N}_i^{(l)}, \tag{6}$$

where $\underline{u}_k^L, \underline{u}_k^R \in \mathbb{C}^{N_c \times 1}$ are implicit functions of τ_k. In (6), the complex magnitude of each user, β_k, has been lumped into the transmitted bits $b_{k,i}$ and $b_{k,i-1}$ to yield $\tilde{\beta}_{k,i}$ and $\tilde{\beta}_{k,i-1}$ respectively. In the second equality in (6) we have $\tilde{\underline{u}}_k = \underline{u}_{(k+1)/2}^L$ and $d_{k,i} = \tilde{\beta}_{k,i-1}$ if k is odd and $\tilde{\underline{u}}_k = \underline{u}_{k/2}^R$ and $d_{k,i} = \tilde{\beta}_{k,i}$ if k is even. $\underline{N}_i^{(l)}$ is a zero mean Gaussian random vector uncorrelated in time.

We can now write $\underline{r}_i^{(l)}$ in vector notation as

$$\underline{r}_i^{(l)} = \mathbf{A}(\underline{\tau}_0)\mathbf{\Phi}^{l-1}\underline{d}_i + \underline{N}_i^{(l)}, \tag{7}$$

where $\underline{\tau}_0 = [\tau_1, \ldots, \tau_k]^\top$, $\mathbf{A}(\underline{\tau}_0) = [\tilde{\underline{u}}_1, \ldots, \tilde{\underline{u}}_{2K}]$, $\underline{d}_i = [d_{1,i}, \ldots, d_{2K,i}]^\top$ and

$$\mathbf{\Phi}^{2K\times 2K} = \begin{bmatrix} \tilde{\phi}_1\mathbf{I}^{(2\times 2)} & & & \\ & \tilde{\phi}_2\mathbf{I}^{(2\times 2)} & & \\ & & \ddots & \\ & & & \tilde{\phi}_k\mathbf{I}^{(2\times 2)} \end{bmatrix}.$$

The observation vector at time iT, $\underline{r}_i \in \mathbb{C}^{MN_c \times 1}$, is then generated by concatenating the $\underline{r}_i^{(l)}$'s for each l as $\underline{r}_i = [\underline{r}_i^{(1)^\top}, \underline{r}_i^{(2)^\top}, \ldots, \underline{r}_i^{(M)^\top}]^\top$. As in (7) we can write $\underline{r}_i$ in vector format as

$$\underline{r}_i = \mathcal{A}(\underline{\tau}_0, \underline{\phi}_0)\underline{d}_i + \underline{N}_i, \tag{8}$$

where $\underline{N}_i^{MN_c \times 1} = \left[\underline{N}_i^{(1)^\top}, \ldots, \underline{N}_i^{(M)^\top}\right]^\top$ is a zero-mean, white, circular, complex Gaussian noise vector, $\underline{\phi}_0 = [\phi_1, \ldots, \phi_K]^\top$ and

$$\mathcal{A} = \begin{bmatrix} \mathbf{A} \\ \mathbf{A\Phi} \\ \vdots \\ \mathbf{A\Phi}^{M-1} \end{bmatrix}.$$

As seen above, we are faced with the problem of estimating $\underline{\Theta}_0 = (\underline{\tau}_0, \underline{\phi}_0)$ from the observations $\underline{r}_i$.

3 Subspace Based Estimation of Signal Parameters

With the assumptions made on the noise matrix $\underline{N}_L$, we can derive the correlation matrix of the observations,

$$\mathbf{R} = E[\underline{r}_i\, \underline{r}_i^\dagger] = \mathcal{A}\mathbf{D}\mathcal{A}^\dagger + \sigma^2\mathbf{I}, \tag{9}$$

where $\mathbf{D} = E[\underline{d}_i\, \underline{d}_i^\dagger] \in \mathbb{C}^{2K \times 2K}$ is a diagonal matrix. If $\mathcal{A}\mathbf{D}$ is of full rank $2K$, the $MN_c - 2K$ smallest eigenvalues of $\mathbf{R}$ are all equal to σ^2. We can form the eigenvalue decomposition of $\mathbf{R}$ as

$$\mathbf{R} = \mathbf{U\Sigma U}^\dagger = \mathbf{E_s\Lambda_s E_s}^\dagger + \mathbf{E_n\Lambda_n E_n}^\dagger = \mathbf{E_s}\tilde{\mathbf{\Lambda}}_\mathbf{s}\mathbf{E_s}^\dagger + \sigma^2\mathbf{I}, \tag{10}$$

where $\mathbf{E_n} = \sigma^2\mathbf{I}$, $\tilde{\mathbf{\Lambda}}_\mathbf{s} = \mathbf{\Lambda_s} - \sigma^2\mathbf{I}$ and $\mathbf{U} = [\mathbf{E_s}\ \mathbf{E_n}]$ is a unitary matrix. The columns of $\mathbf{E_s} \in \mathbb{C}^{MN_c \times 2K}$ span the *signal subspace* and likewise, the columns of $\mathbf{E_n} \in \mathbb{C}^{MN_c \times MN_c - 2K}$ span the orthogonal *noise subspace* [12, 13]. From (9) and (10) we see that the column spaces of $\mathbf{E_s}$ and $\mathcal{A}(\underline{\Theta}_0)$ are the same, assuming that $\mathbf{D}$ is of full rank. Therefore the signal vectors, $\breve{\underline{u}}_k^{R(L)}$, which are the columns of $\mathcal{A}(\underline{\Theta}_0)$ are orthogonal to the noise subspace, $\mathbf{E_n}$ (Note here that if $\breve{\underline{u}}_k^R$ and $\breve{\underline{u}}_k^L$ are orthogonal to $\mathbf{E_n}$, so is $\breve{\underline{u}}_k = \breve{\underline{u}}_k^R + \breve{\underline{u}}_k^L$ [4]). This suggests a method of estimating $\underline{\Theta}_0$: find the $\underline{\Theta}_0$ such that $\{\breve{\underline{u}}_k\}_{k=1}^{2K}$ are orthogonal to $\mathbf{E_n}$.

In reality we do not have access to $\mathbf{R}$, but only its estimate $\hat{\mathbf{R}}$ given by $\hat{\mathbf{R}} = 1/L \sum_{i=1}^{L} \underline{r}_i\, \underline{r}_i^\dagger$, which has an eigenvalue decomposition

$$\hat{\mathbf{R}} = \hat{\mathbf{E}}_\mathbf{s}\hat{\mathbf{\Lambda}}_\mathbf{s}\hat{\mathbf{E}}_\mathbf{s}^\dagger + \hat{\mathbf{E}}_\mathbf{n}\hat{\mathbf{\Lambda}}_\mathbf{n}\hat{\mathbf{E}}_\mathbf{n}^\dagger, \tag{11}$$

and hence in general, no value of $\underline{\Theta}_0$ will render the signal vectors orthogonal to $\mathbf{E_n}$. However we can minimize the projection of $\mathcal{A}(\underline{\Theta}_0)$ into the noise subspace and pose the following problem

$$\hat{\underline{\Theta}}_0 = \arg \min_{\underline{\Theta}_0 = (\underline{\tau}_0, \underline{\phi}_0)} \mathrm{Tr}\left(\mathbf{P}_\mathcal{A}(\underline{\Theta}_0)\hat{\mathbf{E}}_\mathbf{n}\hat{\mathbf{E}}_\mathbf{n}^\dagger\right),$$

where $\mathrm{Tr}(\cdot)$ is the trace of a matrix and $\mathbf{P}_{\mathcal{A}} = \mathcal{A}\mathcal{A}^{+}$ is the projection matrix onto the column space of $\mathcal{A}$ ($\mathcal{A}^{+}$ is the pseudoinverse of $\mathcal{A}$). This algorithm, called the multidimensional MUSIC [12], still involves a $2K$-dimensional optimization. The MUSIC algorithm is a suboptimal version of this algorithm, where the problem is reduced to K two-dimensional searches (for τ and ϕ). The problem can be posed as

$$[\hat{\tau}, \hat{\phi}] = \arg\max_{\tau,\phi} \frac{1}{\mathrm{Tr}(\mathbf{P}_{\underline{\breve{u}}_k}(\tau,\phi)\hat{\mathbf{E}}_{\mathbf{n}}\hat{\mathbf{E}}_{\mathbf{n}}^{\dagger})} = \arg\max_{\tau,\phi} \frac{1}{\|\underline{\breve{u}}_k^{\dagger}(\tau,\phi)\hat{\mathbf{E}}_{\mathbf{n}}\|_2^2}. \tag{12}$$

This is equivalent to minimizing $\|\underline{\breve{u}}_k^{\dagger}(\tau,\phi)\hat{\mathbf{E}}_{\mathbf{n}}\|_2^2$ and the MUSIC problem formulation reduces to

$$[\hat{\tau}, \hat{\phi}] = \arg\min_{\tau,\phi} \underline{u}_k^{\dagger} \left(\sum_{l,m=1}^{M} \hat{\mathbf{E}}_l \hat{\mathbf{E}}_m^{\dagger} e^{j(m-l)\phi} \right) \underline{u}_k \,, \tag{13}$$

where $\underline{u}_k = \underline{u}_k^R + \underline{u}_k^L$ and $\hat{\mathbf{E}}_l \in \mathbb{C}^{N_c \times MN_c - 2K}$ is composed of the l^{th} set of N_c rows of $\hat{\mathbf{E}}_{\mathbf{n}}$.

4 Results and Future Directions

Preliminary investigations into the performance of the MUSIC algorithm have proved promising. The problem as posed in (13) has not yet afforded an elegant closed-from solution and hence a nonlinear two dimensional optimization has to be performed.

An immediate benefit from using antenna arrays is the substantial increase in the dimension of the noise subspace. This subspace, now spanned by the columns of $\mathbf{E}_{\mathbf{n}}$, has dimension $MN_c - 2K$, where K is the number of signals trying to be acquired. The number of users K, that can be supported by the system (i.e., whose timing and DOA parameters can be thoretically estimated) is limited by the dimension of the noise subspace and is upper bounded by $MN_c/2$. These conclusions are corroborated in the accompanying plots.

Figures 3 and 4 show graphs of the rmse of the timing and DOA estimates for increasing number of users on the channel, again conditioned on acquisition having occurred. As we see, a single user channel would not have been able to support more than 3 users at a given time. Figure 5 is a graph of the probability of acquisition as a function of the number of users on the channel, for different values of the window size, $L = 200, 300$. The SNR in the above plots was 8dB.

Figure 6 shows a plot of the surface of the MUSIC metric calculated from (13) for $N_c = 7$ and $K = 9$ and finally, Figure 7 is a similar plot for the case where the code-length is 31 with 33 users on the channel. As can be seen from the positions of the peaks, the timing and DOA estimates are very accurate. The SNR was kept at 8dB and the window length at 200 for both plots.

Further issues to be considered are the estimation of the users' amplitude and phases and the detection of their bit streams. The authors are investigating the use of decorrelating detectors for the detection of the users' bits.

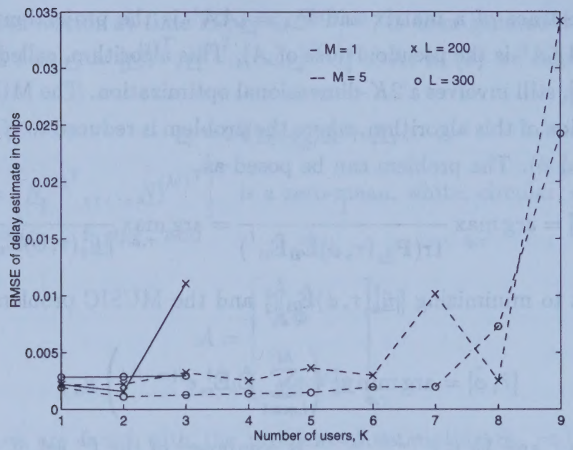


Figure 3: RMSE of delay estimate for MUSIC algorithm with increasing number of user K [$N_c = 7$, SNR = 8dB, MAI = 0dB]

References

[1] U. Fawer and B. Aazhang, "A multiuser receiver for code division multiple access communications over multipath channels," in *Proceedings of GLOBCOM (Communication Theory Mini-Conference)*, Dec. 1992.

[2] S. Y. Miller and S. C. Schwartz, "Parameter estimation for asynchronous multiuser communication," in *Proceedings of the 1989 Conference on Information Sciences and Systems*, pp. 294–299, 1989.

[3] S. E. Bensley, "A multiple-cell search technique for the aquisition of PN sequences," tech. rep., Rice University, Spring 1993.

[4] S. E. Bensley and B. Aazhang, "Subspace-based estimation of multipath channel parameters for code division multiple access communication systems marmar," in *GLOBCOM*, 1994. Submitted for presentation.

[5] E. G. Ström, S. Parkvall, and B. E. Ottersen, "Near-far resistant propagation delay estimators for asynchronous direct-sequence code division multiple access systems," in *Proceedings of the 1994 International Zurich Seminar on Digital Communications*, pp. 251–260, Springer Lecture Notes in Computer Science 783, Mar. 1994.

[6] D. Dlugos and R. Scholtz, "Aquisition of spread spectrum signals by an adaptive array," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 37, pp. 1253–1270, August 1989.

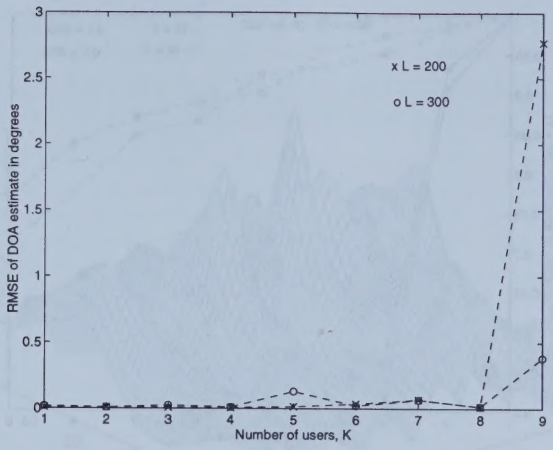


Figure 4: RMSE of angle estimate for MUSIC algorithm with increasing number of user K [$N_c = 7$, SNR = 8dB, MAI = 0dB]

[7] L. E. Brennan and I. S. Reed, "An adaptive signal processing algorithm for communications," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-18, pp. 124–130, 1982.

[8] B. Ottersten, M. Viberg, P. Stoica, and A. Nehorai, "Exact and large sample maximum likelihood techniques for parameter estimation and detection in array processing," in *Radar Array Processing* (T. J. S. S. haykin, J. Litva, ed.), vol. 25 of *Springer Series in Information Sciences*, Springer-Verlag, 1993.

[9] P. Stoica and A. Nehorai, "Statistical efficiency study of direction estimation methods part I: Analysis of MUSIC and preliminary study of MLM," in *Advances in Spectrum Analysis and Array Processing — Volume II* (S. Haykin, ed.), pp. 263–305, Englewood Cliffs, New Jersey: Prentice Hall, 1991.

[10] D. H. Johnson and D. E. Dudgeon, *Array Signal Processing: Concepts and Techniques.* Englewood Cliffs, NJ: Prentice Hall, 1993.

[11] P. Stoica and K. Sharman, "Maximum likelihood methods for direction-of-arrival estimation," *IEEE Trans. ASSP*, vol. 38, pp. 1132–1143, 1990.

[12] A. Van der Veen, E. Deprettere, and A. Swindlehurst, "Subspace-based signal analysis using singular value decomposition," *Proceedings of the IEEE*, vol. 81, pp. 1277–1308, Sept. 1993.

[13] P. Stoica and A. Nehorai, "Performance comparison of subspace rotation and music methods for direction estimation," *IEEE Trans. SP*, vol. 39, pp. 446–453, 1991.

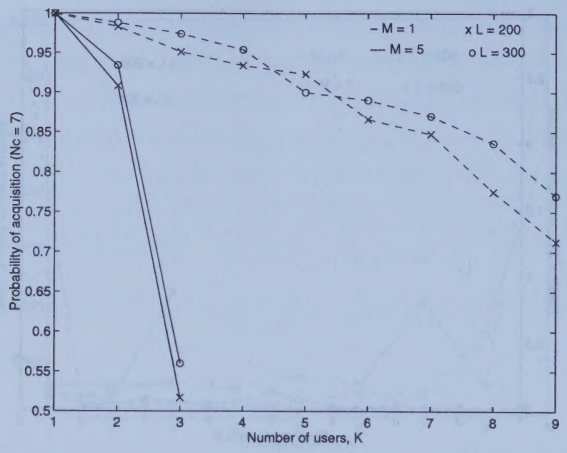


Figure 5: Probability of acquisition for a 5-element antenna array with increasing number of user K [$N_c = 7$, SNR = 8dB, MAI = 0dB]

[14] A. F. Naguib and A. Paulraj, "Performance of cdma cellular networks with base-station antenna arrays," in *Mobile Communications - Advanced Systems and Components*, Springer-Verlag, 1994.

[15] B. Khalaj, A. Paulraj, and T. Kailath, "Antenna arrays for cdma systems with multipath," in *Proc. Milcomm'93*, (Bedford, MA), November 1993.

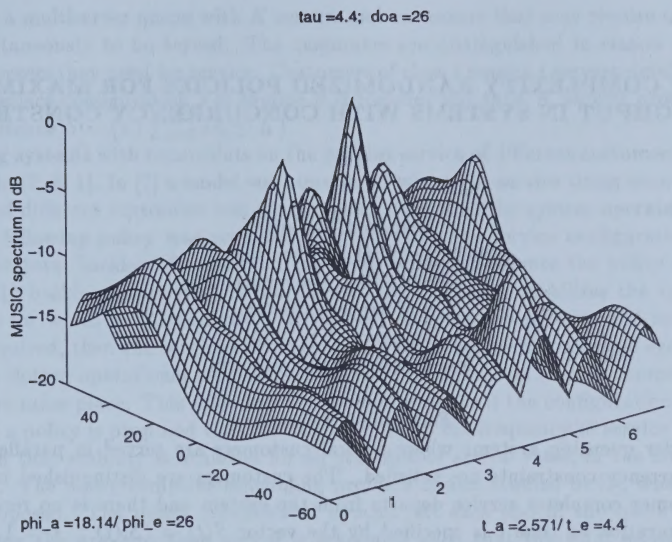


Figure 6: Surface of MUSIC metric for $K = 9$ users with $N_c = 7$ [SNR = 8dB, MAI = 0dB]

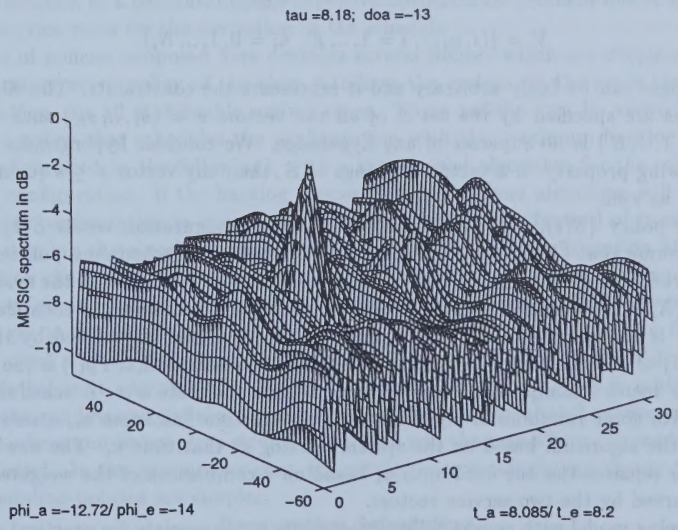


Figure 7: Surface of MUSIC metric for $K = 33$ users with $N_c = 31$ [SNR = 8dB, MAI = 0dB]

LOW COMPLEXITY RANDOMIZED POLICIES FOR MAXIMUM THROUGHPUT IN SYSTEMS WITH CONCURRENCY CONSTRAINTS

Leandros Tassiulas*
Electrical Engineering
Polytechnic University
6 Metrotech Center
Brooklyn, NY 11201
leandros@olympos.poly.edu

Extended Summary

We consider queueing systems where several customers are served in parallel as long as certain concurrency constraints are satisfied. The customers are distinguished in K classes. After a customer completes service departs from the system and there is no rerouting. The service configuration at time t is specified by the vector $S(t) = (S_i(t) : i = 1, .., K)$. The number of class i customers receiving service at time t will be equal to $S_i(t)$, when there are more than $S_i(t)$ class i customers in the system at time t, and equal to the number of class i customers otherwise. There is a maximum number N_i of class i customers, that may be served simultaneously. In addition there are cross-class constraints on the parallel service expressed in terms of an undirected hypergraph $G = (V, E)$, called the *constrain hypergraph*. Hypergraph is the generalization of a graph where an edge can be any collection of nodes and not just a pair of nodes [2]. The set of nodes of the constrain hypergraph is

$$V = \{(i, n_i) : \ i = 1, .., K, \ n_i = 0, 1, .., N_i\}.$$

The set of edges can be fairly arbitrary and it represents the constraints. The eligible service configurations are specified by the set $\mathcal{S}$ of all the vectors $s = (s_1, .., s_K)$ such that the set $\{(i, s_i) : i = 1, .., K\}$ is no superset of any hyperedge. We consider hypergraphs such that $\mathcal{S}$ has the following property: if a vector s belongs to $\mathcal{S}$, then any vector $s' \leq s$ (componentwise) belongs to $\mathcal{S}$ as well.

A service policy $\{S(t), \ t \geq 0\}$ specifies the service configuration vector $S(t)$ at all times $t \geq 0$. We assume that $\{S(t), \ t \geq 0\}$ is piecewise constant. The scheduling policies considered here rely on randomized algorithms that compute a service vector based on the system backlog. Let $X(t) = (X_i(t) : i = 1, .., K)$ be the vector of backlogged customers of each class at time t and $\mathcal{X} = R^K$ is the space where it lies. A randomized algorithm is specified by the collection $\{P_x(\cdot), x \in \mathcal{X}\}$ of distributions $P_x(\cdot)$ on $\mathcal{S}$, with the interpretation that $P_x(\cdot)$ is the distribution of the service vector s computed by the algorithm when the state is x. A scheduling policy is associated with some randomized algorithm. At certain time instances u_i, a service vector is obtained by the algorithm based on the system backlog at that time u_i. The newly computed service vector replaces the one used until u_i based on a comparison of the weighted backlog of the classes served by the two service vectors.

The queueing model with compatibility constraints is appropriate for practical systems that cannot be modeled by traditional queueing networks with distinct servers. For specific applications, ranging from database concurrency control and parallel processing, to radio networks, the reader is referred to [7, 6, 1, 4, 5]. An example of such a system that we will refer to in the

*This research was supported in part by NSF under grant NCR-9211417 and by the New York State Center for Advanced Technology in Telecommunications, Polytechnic University.

following, is a multiserver queue with K servers and customers that may require more than one server simultaneously to be served. The customers are distinguished in classes based on the number of servers they need for service. Customers of class i require i servers simultaneously for service. A service configuration s is eligible if it requires less than K servers providing service in parallel. Hence $\mathcal{S} = \{s : \sum_{i=1}^{K} is_i \leq K\}$.

Queueing systems with constraints on the parallel service of different customers were considered before in [7, 6, 1]. In [7] a model was considered where the service times were constant and the service of different customers was synchronized such that the system operates in a slotted mode. The following policy was proposed. At every slot the service configuration is selected such that the total backlog of the served queues is maximum. Hence the policy gives priority to the heavily backlogged queues. It was shown that the policy stabilizes the system for any arrival rates for which it is stabilizable. If the service times are random and non-preemptive service is required, then the service completions are not synchronized and the system does not maintain its slotted operational mode. As the service configuration changes, some forced idling of the servers takes place. This may degrade the performance if the configuration changes very often. In [6] a policy is proposed that controls adaptively the frequency of service configuration changes such that stability is achieved for all stabilizable arrival rates, in the non-preemptive case as well. The non-preemptive scheduling problem is also studied in [1], where a different policy is proposed. The policy schedules at epochs. At each epoch, the batch of customers arrived during the previous time period is scheduled such that the total service completion time is minimized asymptotically. That policy achieves maximum throughput for stationary arrival and service processes.

The above policies achieve maximum throughput but they have certain weaknesses. The determination of the service configuration for which the aggregate backlog of the served queues is maximized, a step required by the policies in [7, 6], is a computationally demanding task and, depending on the system, it may be NP-hard in some cases. The policy proposed in [1] involves the solution of a computationally intensive optimization problem and it also needs the arrival and service rates for the derivation of the schedule.

The class of policies proposed here contains several polices which are simple computationally. Furthermore every policy of the class stabilizes the system, in the sense that it achieves bounded backlog, for all stabilizable arrival rates. These policies can be viewed as a combination of the policy that schedules the configuration with the maximum backlog at each step (optimal configuration in the following), with a randomized algorithm for the computation of the optimal configuration. If the backlog freezes, the randomized algorithm will compute the optimal service configuration in a simulated annealing fashion [3]. Instead of freezing the state though and running the algorithm until it computes the optimal configuration at every epoch of configuration change, we let it run in parallel with the system operation, doing one iteration (random selection and update) every time the configuration changes. Since the backlog of the system changes as the algorithm runs, there is no guarantee that at any time the configuration that is used will be the optimal (recall that the optimal configuration depends on the state). Nevertheless it approximates the optimal close enough such that it achieves stability. In addition the randomized policies are not depended of the arrival and service rates.

The case where non-preemptive service is required and the case where preemption is allowed are distinguished. In the preemptive case, the service vector can change at any time instance and the scheduling policies are simpler.

Preemptive Scheduling

Assume that the customers of each class arrive according to independent Poisson processes with rate λ_i for class i. The service times are independent and exponentially distributed with rate μ_i for class i. Consider scheduling policies $\{S(t), t \geq 0\}$ which are ergodic and the queue length process under those is ergodic as well. The necessary condition for stability, in the sense that the arrival rate is equal to the departure rate for each class, under those policies is the

following

$$(\lambda_i/\mu_i, i = 1, .., K) \in co(\mathcal{S}). \quad (1)$$

A proof of the necessity of (1) under similar conditions is contained in [7, 6] and it is only outlined here. Let $D_i(t)$ be the total number of class i departures until time t. The departure process of each class i is modulated Poisson with rate at time t equal to $\min\{X_i(t), S_i(t)\}$. We have

$$\lim_{t\to\infty} \frac{1}{t} D(t) = \lim_{t\to\infty} \frac{1}{t} \mu \otimes \int_0^t \min\{X(\tau), S(\tau)\} d\tau$$

where $D(t) = (D_i(t) : i = 1, .., K)$ and by definition, the operation $\otimes$ between two vectors is their pointwise product. Notice that $\frac{1}{t}\int_0^t \min\{X(\tau), S(\tau)\} d\tau$ belongs to $co(\mathcal{S})$ for every time t, therefore the necessity of (1) follows.

Consider scheduling policies $\{S(t), t \geq 0\}$ specified using the following three entities: a randomized algorithm $\{P_x(\cdot), x \in \mathcal{X}\}$, a sequence of update times $\{u_n\}_{n=1}^{\infty}$, and an update rule. The randomized algorithm and the update rule determine a sequence $\{S^n\}_{n=1}^{\infty}$ of random service vectors. The scheduling policy then is

$$S(t) = S^n \text{ if } u_n \leq t < u_{n+1}.$$

The sequence of service vectors is derived as follows. The randomized algorithm is applied on the state of the system at times $\{u_n\}_{n=1}^{\infty}$ and derives a sequence of independent random service vectors $\{\hat{S}^n\}_{n=1}^{\infty}$, where $\hat{S}^n$ is distributed according to $P_{X(u_n)}$. The update rule derives the sequence $\{S^n\}_{n=1}^{\infty}$ as follows

$$S^n = \begin{cases} \hat{S}^n, & \text{if } X^T(u_n)\hat{S}^n > X^T(u_n)S_{n-1}, \\ S_{n-1}, & \text{otherwise,} \end{cases}$$

and S_0 is arbitrary.

Consider the class of policies Π such that the sequence of update times is Poisson and the randomized algorithm satisfies the following property:

C. There exists $\epsilon > 0$ such that

$$P_x(\arg\max_{s\in\mathcal{S}}\{x^T s\}) \geq \epsilon, \ \forall x \in \mathcal{X}.$$

Property C requires that with positive probability, for any state x, a schedule s that maximizes $x^T s$ can be computed by the randomized algorithm.

The class Π is fairly wide, and contains some policies with fairly low computational complexity. Consider for instance the following algorithm for selecting the vectors $\hat{S}^n$

Select each component S_i of vector $S = (S_i : i = 1, .., K)$ randomly with uniform distribution on $\{0, 1, .., N_i\}$. If S belongs to $\mathcal{S}$ then let $\hat{S}^n = S$, otherwise let $\hat{S}^n = 0$.

The above algorithm for selecting a schedule vector is clearly of low computational complexity. Furthermore the probability of selecting any vector in $\mathcal{S}$ different than 0, is equal to $\Pi_{i=1}^K (N_i)^{-1}$, independently of the system state. Therefore property C holds with $\epsilon = \Pi_{i=1}^K (N_i)^{-1}$. The policies considered in [7, 6] are based on a deterministic algorithm that computes for every state x a service vector s such that $x^T s$ is maximum. Therefore property C holds in that case with $\epsilon = 1$ and obviously those policies belong to Π as well.

Theorem 1 *Under any policy in Π, and if the vector $(\lambda_i/\mu_i : i = 1, .., K)$ belongs to the interior of $co(\mathcal{S})$, the system is stable in the sense that the Markov chain $\{(X(u_n), S^n)\}_{n=1}^{\infty}$ has a stationary distribution under which $E[X(u_n)] < \infty$.*

The proof is based on Lyapunov functions.

Non-Preemptive Scheduling

If non-preemptive service is required, then every time the service vector should change, some service capacity will be wasted. Either the customers which are in the middle of their service will be interrupted, in which case the service already provided will be wasted, or some service capacity will be wasted waiting for the last customers of the previous service vector to finish service. Hence the transition from one service vector to another should be done rare enough to avoid significant under utilization of the servers. Also the transition from one service vector to another should be done as gracefully as possible.

Similarly to the preemptive case, a randomized algorithm that satisfies condition C, is applied on the state of the system at the update times $\{u_n\}_{n=1}^{\infty}$ and derives a sequence of independent random service vectors $\{\hat{S}^n\}_{n=1}^{\infty}$, where $\hat{S}^n$ is distributed according to $P_{X(u_n)}$. The policy is dictated by a sequence $\{S^n\}_{n=1}^{\infty}$ of random service vectors that is derived by $\{\hat{S}^n\}_{n=1}^{\infty}$ based on the following update rule

$$S^n = \begin{cases} \hat{S}^n, & \text{if } X^T(u_n)\hat{S}^n > \alpha(X^T(u_n))X^T(u_n)S_{n-1}, \\ S_{n-1}, & \text{otherwise,} \end{cases}$$

where $\alpha(x) = 1 + |x|^{-\delta}$. The initial service vector S_0 can be arbitrary. The scheduling policy $\{S(t), t \geq 0\}$ is as follows

$$S(t) = S^n \text{ if } u_n \leq t < u_{n+1}.$$

At the times of the switching of the service vector, the transition to the new service vector is done gracefully, without interrupting an ongoing service, as follows. If at the time u_n of the switching of the service vector, the number of class i customers under service is higher than S_i^n, the number specified by the new service vector, then customers of class i completing service are not replaced until the number settles below S_i^n. At the time of a service completion or a new arrival, if the number of served customers of some class i can increase without exceeding S_i^n and without violating the service constraints, then it does so. Under the class of policies specified above the network is stable when non-preemptive service is required, in the sense stated in theorem 1.

References

[1] N. Bambos and J. Walrand. Scheduling and stability aspects of a general class of parallel processing systems. *Adv. Appl. Prob.*, 25:176–202, 1993.

[2] C. Berge. *Graphs and Hypergraphs.* North-Holland, 1973.

[3] B. Hajek. Cooling schedules for optimal annealing. *Mathematics of Operations Research*, 13(2):311–329, 1988.

[4] D. Mitra. Probabilistic models and asymptotic results for concurrent processing with exclusive and nonexclusive locks. *SIAM Jour. Comput.*, 14:1030–1051, 1985.

[5] D. Mitra and P.J. Weinberger. Probabilistic models of database locking: solutions computational algorithms and asymptotics. *Journal of ACM*, 31:855–878, 1984.

[6] L. Tassiulas and P. Bhattacharya. Allocation of interdependent resources for maximum throughput. *IBM T. J. Watson Research Center*, Technical Report, 1993.

[7] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Trans. on Automatic Control*, 37(12):1936–1948, December 1992.

The Baccelli-Foss Saturation Rule for stability for continuous time input processes

Eitan Altman
INRIA Sophia Antipolis, B.P. 93
06902 Sophia Antipolis, France

Abstract

We analyze the stability of systems that are driven by input processes that are right continuous and have left limits. These include stochastic fluid models frequently used in telecommunications for modeling variable rate inputs, as well as jump processes. Our approach is an adaptation of the Saturation Rule developed recently by Baccelli and Foss. The analysis is carried under the assumption that the input process is stationary ergodic, or that it has some stationary periodic structure.

Keywords: Saturation Rule, stationarity, Ergodicity.

1 Introduction

A very powerful and elegant new approach for studying the stability of non-Markovian stochastic systems has been recently developed by Baccelli and Foss in [4]. It was very successfully applied to generalized Jackson networks [3], stochastic Petri nets [5], and polling systems [7]. This method considers systems as operating on some exogenous marked point processes. For example, in queueing applications, the times at which jumps occur may correspond to arrival of customers, and the marks could correspond to service times and routing information. One then focuses on the daters, which are random variables that indicate the time at which some local or global activity has terminated, given that the input process is restricted to some finite time. The methodology is based on

spearable and monotone axioms that are assumed to be satisfied by the daters. They are shown to imply sub-additivity properties. Under some fairly general stochastic assumptions on the input processes, Baccelli and Foss obtained a first order ergodic theorem, by applying Kingman's sub-additive ergodic theorem, and some refined second-order ergodic theorems.

In this note we extend the above framework to general exogenous input processes that are right continuous and have left limits. This generalization is partly motivated by the growing use of stochastic fluid models in telecommunication and production systems [8, 9]. Our results are based on adaptation of the approach in [4] (see also [2]), and by introducing accelerating and decelerating time change functions, that allows to compare daters that correspond to different processes. An additional monotone axiom needs to be satisfied in our framework, which we call the "dual monotonicity", in order to apply an extention of Kingman's ergodic theorem for continuous time ([6] p. 177). We further obtain stability results for input processes that have some periodic behavior. An example of a G/G/1 type queue is given finally.

2 The Model

We begin by presenting some axioms that refer to the behavior of the system for given fixed input processes $N : \mathbb{R} \to G$, where G is some complete separable metric space, endowed with the standard Borel field $\mathcal{G}$. N is assumed to be right continuous with left limits. Let D be the set of paths which are right continuous with left limits, endowed with the standard Borel σ-Algebra $\mathcal{D}$. N may be typically a monotone nondecreasing function of time; in the case of a G/G/1 queue, it typically takes values in $\mathbb{R}$, and has the interpretation of the cumulative workload arriving to the system. Define

- $\theta : \mathbb{R} \times D \to D$ the operation of shifting by a constant, given by $\theta_s N(t) = N(t+s)$. We shall use (with some abuse of notation) θ also as a shift over sets, i.e. $\theta : \mathbb{R} \times \mathcal{D} \to \mathcal{D}$, where $\theta_s A := \{\theta_s N, N \in A\}$ for any $A \in \mathcal{D}$.
- The operation of scaling N by a constant c is written as cN, i.e. $cN(t) := N(t/c)$.
- $N_{r,s}(t) :=$ is the restriction of N to the interval $(r, s]$. (This means that out side this interval, the input is inactive).

- $\mathcal{T} : \mathbb{R} \to \mathbb{R}$:= the set of monotone functions which are right continuous. Functions in $\mathcal{T}$ will be identified as "time-changes".
- $\Theta : \mathcal{T} \times D \to D$: the operation of time change; given $\pi \in \mathcal{T}$, $N \in D$, we define $\Theta(\pi, N)(t) = N(\pi(t))$.

We say that a time change π is an acceleration if $\pi(t) \geq t$ for all t, and it is deceleration if $\pi(t) \leq t$ for all t. We say that N is slower than N' (or N' is faster than N), and denote it by $N \prec N'$, if there exists some acceleration π such that $N' = \Theta(\pi, N)$. Finally, we note that

$$c(N_{s,t}) = (cN)_{cs,ct} \text{ and } \theta_r(N_{s,t}) = (\theta_r N)_{s-r,t-r}, \tag{1}$$
$$\text{if } N' = \Theta(\pi, N) \text{ then } N'_{r,s}(t) = N_{\pi(r),\pi(s)}(\pi(t)) \text{ and } \Theta(\pi, N_{r,s}) = N'_{\pi^{-1}(r),\pi^{-1}(s)}$$

whenever π^{-1} is well defined.

Following [3], we define the maximal dater $X(N)$ to be the time of the last activity in the network, when it is fed by N (given that the system is empty at $t = -\infty$). X thus describes some property of the system to which process N is fed. Define for any $s \leq t$ $Z_{s,t}(N) := X(N_{s,t}) - t$. We assume that X has the following properties. For all s, t, N, with $s \leq t$:

1. **Causality:** $X(N_{s,t}) \geq t$.
2. **External monotonicity:** $X(N') \geq X(N_{s,t})$ whenever $N' \prec N_{s,t}$.
3. **Homogeneity:** For any $r \in \mathbb{R}$, $X(\theta_r N) = X(N) + r$.
4. **Separability:** For any $r \in \mathbb{R}$ such that $s \leq r < t$, if $X(N_{s,r}) \leq r$, then $X(N_{s,t}) = X(N_{r,t})$.

Lemma 2.1 *Under conditions 1-4, X and $Z_{s,t}$ satisfy the following:*

5. **Internal monotonicity property:** *For all r, s, t with $s \leq r < t$ and all N,*

$$X(N_{s,t}) \geq X(N_{r,t}), \qquad Z_{s,t}(N) \geq Z_{r,t}(N). \tag{2}$$

Proof: The proof is given in [1]. It is based on choosing arbitrary r, s, t with $s \leq r < t$ and N, and then defining the process $N' = \Theta(\pi^1, N_{s,t})$, where the accelerating time

change $\pi^1 \in \mathcal{T}$ is given by

$$\pi^1(t) = \begin{cases} t + Z_{s,r}(N) & \text{if } t < r - Z_{s,r}(N) \\ r & \text{if } r - Z_{s,r}(N) \leq t < r \\ t & \text{if } t \geq r \end{cases} \tag{3}$$

The remaining of the proof follows from ideas similar to those in [4]. ■

Lemma 2.2 *Under conditions 1-4, $Z_{s,t}$ satisfies the following*
6. Sub-additive property: *For all r, s, t with $s \leq r < t$ and all N,*

$$Z_{s,t}(N) \leq Z_{s,r}(N) + Z_{r,t}(N). \tag{4}$$

Proof: The proof is given in [1]. It is based on introducing the following decelerating time change π^2:

$$\pi^2(t) = \begin{cases} t & \text{if } t < r \\ r & \text{if } r \leq t < r + Z_{s,r}(N) \\ t - Z_{s,r}(N) & \text{if } r + Z_{s,r}(N) \leq t \end{cases} \tag{5}$$

and considering the processes $N^1 = \Theta(\pi^1, N)$ and $N^2 = \Theta(\pi^2, N)$. The remaining of the proof follows from ideas similar to those in [4]. ■

3 The stochastic framework

We assume in this section that N is defined on the canonical probability space $(D, \mathcal{D}, P)$, and is stationary ergodic under the shift θ_t. We further assume that it satisfies conditions 1-4 on all trajectories (with probability 1). We need the following additional assumption.

$$E\left[\sup_{0<s\leq 1} Z_{0,s}(N)\right] < \infty. \tag{6}$$

By the internal monotonicity this implies in particular

$$E\left[\sup_{0\leq r<s\leq 1} Z_{r,s}(N)\right] < \infty. \tag{7}$$

Since sup is a convex function, (6) also implies (by Jensen's inequality) that

$$\sup_{0<s\leq 1} EZ_{0,s}(N) < \infty. \tag{8}$$

(8) implies that for any r and s such that $r \leq s$, $EZ_{r,s}(N) < \infty$. Indeed, by expressing s as $s = K\epsilon$ for some $0 < \epsilon \leq 1$ with K being some integer, we have

$$EZ_{r,s}(N) \leq_5 EZ_{0,s}(N) \leq_6 KEZ_{0,\epsilon}(N) < \infty. \tag{9}$$

(7), (9) and the nonnegativity of $Z_{0,t}(N)$ allow us to use a generalization of Kingman's subadditive ergodic theorem ([6] p. 177), which yields

Lemma 3.1 *Assume Conditions 1-4 and (6). Then there exists some finite non-negative constant γ such that*

$$\lim_{t\to\infty} \frac{Z_{-t,0}(N)}{t} = \lim_{t\to\infty} \frac{Z_{0,t}(N)}{t} = \lim_{t\to\infty} \frac{EZ_{0,t}(N)}{t} = \lim_{t\to\infty} \frac{EZ_{-t,0}(N)}{t} = \gamma \tag{10}$$

P-a.s.

Note that Lemma 3.1 implies that $\lim_{t\to\infty} X(N_{0,t})/t = \gamma + 1$. The proof of the following Lemma is the same as in [2] Chapter 9 and [4].

Lemma 3.2 *Assume Conditions 1-4 and (6). Then*
(i) Define $A = \{lim_{t\to\infty} Z_{-t,0}(N) = \infty\}$. Then $P(A)$ is either one or zero.
(ii) $Z_{-ct,0}(cN)$ is decreasing in c and $Z_{0,ct}(cN)$ is increasing in c; for all $c > 0$, there exists a non-negative constant $\gamma(c)$ such that the following hold for the scaled processes:

$$\lim_{t\to\infty} \frac{Z_{-ct,0}(cN)}{t} = \lim_{t\to\infty} \frac{Z_{0,ct}(cN)}{t} = \lim_{t\to\infty} \frac{EZ_{-ct,0}(cN)}{t} = \lim_{t\to\infty} \frac{EZ_{0,ct}(cN)}{t} = \gamma(c), \quad a.s.; \tag{11}$$

$\gamma(c)$ is decreasing in c and $\gamma(c) + c$ is increasing in c. In particular, $\gamma + 1 \geq \gamma(0)$ where $\gamma = \gamma(1)$.

Remark 3.1 *The process cN is not well defined if $c = 0$ (it is not in $\mathcal{D}$ anymore) unless some extra assumptions are made. Therefore, unlike the framework studied in [4], we do not allow $c = 0$ in the above Lemma. However, since $\gamma(c)$ is decreasing in c, the limit $\gamma(0) := \lim_{c\to 0} \gamma(c)$ is well defined. This problem requires special attention in the proof of the second part of the following main result.*

Theorem 3.1 *Assume conditions 1-4 and (6).*
(i) If $\gamma(0) > 1$ *then* $Z_0(N) := \lim_{t\to\infty} Z_{-t,0}(N) = \infty$ *a.s.*
(ii) If $\gamma(0) < 1$ *then* $Z_0(N) < \infty$ *a.s.*

Proof: The proof is given in [1]. The proof of (ii) requires special attention due to Remark 3.1. ■

We thus obtain a stationary ergodic process $Z_t(N) := \theta_t Z_0(N)$ under the assumptions that $\gamma(0) < 1$.

Finally, we present a simple sufficient condition for (6):
7. Dual monotonicity: For any $r \le s \le t$, $X(N_{r,s}) \le X(N_{r,t})$.

Lemma 3.3 *[1] Assume conditions 1-4 and 7, and that* $EZ_{0,1}(N) < \infty$. *Then (6) holds.*

4 The stochastic periodic framework

Assume that there is some period T, such that the input process is stationary ergodic under the shift $\theta'_n := \theta_{nT}$, $n = 0, 1, 2, ...$ (instead of the shift θ_t, $t \in \mathbb{R}$). This includes in particular the case of periodic deterministic input process, with period T. Assume further that the input process satisfies conditions 1-4, as well as the integrability assumption (6). It follows that Lemma 3.1, 3.2 and Theorem 3.1 still hold. The process $Z_t(N) := \theta_t Z_0(N)$ thus obtained, is stationary ergodic under the shift θ'_n, if $\gamma(0) < 1$. Again, The dual monotonicity can be used together with the assumption $EZ_{0,1}(N) < \infty$ as a sufficient condition to establish (6).

5 Example: a G/G/1 type queue

We consider a queue where the server operates at a unit speed. The arrival process N is right continuous with left limits; it is stationary ergodic (with respect to the shift θ_t, $t \in \mathbb{R}$, or with respect to the shift $\theta'_n = \theta_{nT}$, $n \in \mathbb{N}$, where T is the period). $N(t)$ represents the total accumulated work that arrived to the system till time t. The standard

G/G/1 queue is obtained as special case when work arrives at discrete times. N then becomes a jump process which can be described as a marked point processes, where the times of jumps represent the arrival epochs, and the jump sizes (or the marks) represent the service time required by a customer.

Let W_t denote the amount of work in the system. It evolves according to:

$$W_t = \max\left\{\int_s^t (dN(t) - dt), t_0 < s \leq t, W_{t_0} + \int_{t_0}^t (dN(t) - dt), 0\right\}. \tag{12}$$

If we identify an inactive system with an empty system and take $X(N)$ to denote the time till the system becomes inactive, then it turns out that $X(N)$ does not satisfy the causality condition, since the system may be empty even if positive amount of work arrives to the system at any time, provided that the input rate is smaller than 1. However, if we identify an inactive system as an empty system with idle server then conditions 1-4 and 7 hold. We note that $Z_t(N)$ is simply the work present at the system at time t, so that our results provide the stability conditions (existence of a stationary regime) for the work in the system.

It follows from the monotonicity property in Lemma 3.2 (ii), that

$$\begin{aligned} \gamma(0) &= \lim_{c\to 0}\lim_{t\to\infty} \frac{EZ_{-ct,0}(cN)}{t} = \lim_{c\to 0}\lim_{t\to\infty} \frac{EZ_{0,ct}(cN)}{t} \\ &= \lim_{t\to\infty}\lim_{c\to 0} \frac{EZ_{-ct,0}(cN)}{t} = \lim_{t\to\infty}\lim_{c\to 0} \frac{EZ_{0,ct}(cN)}{t}. \end{aligned} \tag{13}$$

It follows from (12) that $\lim_{c\to 0} Z_{-ct,0}(cN) = \int_{-t}^0 dN(s)$, so (13) implies that $\gamma(0)$ is the average rate at which work arrives to the system per unit of time. More precisely, in the stochastic framework $\gamma(0)$ is the a.s. limit $\gamma(0) = \lim_{t\to\infty} t^{-1}\int_{-t}^0 dN(s)$. For the special case that N is deterministic with period T, it is given by

$$\gamma(0) = \lim_{t\to\infty} t^{-1}\int_{-t}^0 dN(s) = T^{-1}\int_{-T}^0 dN(s). \tag{14}$$

References

[1] E. Altman, "On the Baccelli-Foss Saturation Rule for stochastic stability", manuscript.

[2] F. Baccelli and P. Brémaud, *Elements of Queueing theory*, Springer-Verlag, 1994.

[3] F. Baccelli, and S. Foss, "Stability of Jackson-type Queueing Networks, I", INRIA Report n. 1845, To appear in QUESTA.

[4] F. Baccelli and S. Foss, "On the Saturation Rule for the Stability of Queues", INRIA Report n. 2015, 1993.

[5] F. Baccelli and B. Gaujal, "Stationary Regime and Stability of Free-Choice Petri Nets", *Proc. 11th Int. Conf. on Analysis and Optimization of systems.*

[6] R. Carmona and J. Lacroix, *Spectral Theory of Random Scrœdinger Operators*, Birkhäuser, 1990.

[7] N. Chernova and S. Foss, "On the boundedness of queue-length process in polling systems", In preparation.

[8] D. Mitra, "Stochastic theory of a fluid model of producers and consumers coupled by a buffer", *Adv. of Appl. Prob.* **20**, pp. 646-676, 1988.

[9] T. E. Stern and A. I. Elwalid, "Analysis of separable Markov-modulated rate models for information-handling systems", *Adv. of Appl. Prob.* **23**, pp. 105-139, 1991.

Linear Switching Curve Control of Unreliable Manufacturing Systems*

James R. Perkins† and R. Srikant‡

Abstract

We consider a general fluid flow model, which represents an unreliable manufacturing system consisting of two machine states and two buffer level states. We show that the steady-state probability densities of the buffer levels satisfy a set of coupled hyperbolic partial differential equations. Using Laplace transform techniques, we explicitly determine the steady-state densities. We then discuss the application of these results to the determination of an optimal linear switching curve control for the case where two part-types are produced on a single unreliable machine.

1 Fluid Flow Model

In this paper, we examine a general fluid flow model for an unreliable manufacturing system. The system has two machine states, a "down" state and an "up" state, denoted by 0 and 1, respectively. We assume that the machine state process is modeled as a continuous-time Markov chain, with q_u being the constant rate of transition from State 0 to State 1, and q_d being the constant rate of transition from State 1 to State 0.

Let $x(t) \geq 0$ and $y(t) \geq 0$ be the levels of the two system buffers at time t. Consider the buffer level flows shown in Figure 1, where a, b, c, and d are positive constants. Thus, in State 1, if $x(t) > 0$, then $x(t)$ will decrease at rate b and $y(t)$ will increase at rate a. If $x(t) = 0$ and $y(t) > 0$, then $x(t)$ will remain at 0 and $y(t)$ will decrease at rate d. If $x(t)$ and $y(t)$ are both equal to zero, they will remain there. Similarly, in State 0, $x(t)$ will always decrease at rate c, while $y(t)$ remains unchanged.

Assuming that the capacity condition

$$b\,d\,q_u > c\,(a+d)\,q_d \tag{1}$$

*Supported in part by the National Science Foundation under Grant ECS-9410242.

†Department of Manufacturing Engineering, Boston University, 44 Cummington St., Boston, MA 02215.

‡AT&T Bell Labs. 2C-411, 101 Crawfords Corner Road, Holmdel, NJ 07733.

is satisfied, there will be two two-dimensional steady-state probability densities, $p_1(x,y)$ and $p_0(x,y)$, two one-dimensional steady-state probability densities, $w(y)$ and $s(x)$, and a point mass γ, located at $(0,0)$, as shown in Figure 1.

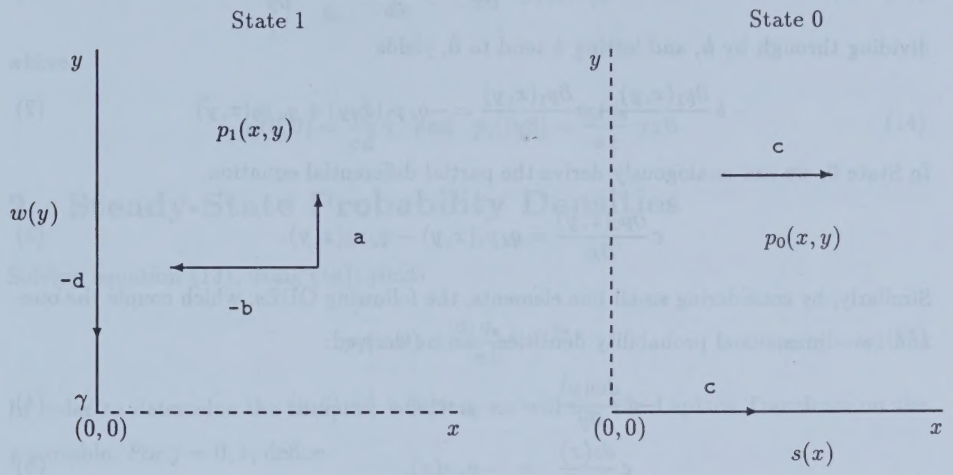


Figure 1: Flows and densities for (x,y) system.

We now derive the partial differential equations that are satisfied by the steady-state probability densities. Disregarding terms which are $o(h)$, there are two possible ways to reach a small interior volume element $dx\,dy$ in State 1, in a small amount of time h:

1. Start in State 1 and remain there for time h.

2. Start in State 0 and, at time t, switch to State 1 and remain there for time $t-h$.

Thus, from Figure 1, using I(·) to denote the indicator function, we have

$p_1(x,y)\,dx\,dy = \iint p_1(u,v)\,(1-q_d\,h)\,I(u-b\,h \in (x,x+dx))\,I(v+a\,h \in (y,y+dy))\,du\,dv +$
$\int_0^h \iint p_0(u,v)\,(1-q_u\,t)\,q_u\,(1-q_d\,(h-t))\,I(u+c\,t-b\,(h-t) \in (x,x+dx))$
$I(v+a\,(h-t) \in (y,y+dy))\,du\,dv\,dt + o(h)$

which equals

$\int_{x+b\,h}^{x+dx+b\,h} \int_{y-a\,h}^{y+dy-a\,h} p_1(u,v)\,(1-q_d\,h)\,du\,dv + \int_0^h \int_{x+b\,h-(b+c)\,t}^{x+dx+b\,h-(b+c)\,t} \int_{y-a\,h+a\,t}^{y+dy-a\,h+a\,t} p_0(u,v)$
$(1-q_u\,t)\,q_u\,(1-q_d\,(h-t))\,du\,dv\,dt + o(h).$

Simplifying, leads to the equation

$$p_1(x,y)\,dx\,dy \;=\; p_1(x+b\,h, y-c\,h)\,(1-q_d\,h)\,dx\,dy +$$

$$\int_0^h p_0(x + b\,h - (b+c)\,t, y - a\,h + a\,t)\,(1 - q_u\,t)\,q_u\,(1 - q_d\,(h-t))\,dx\,dy\,dt + o(h).$$

Using the Taylor series expansion

$$p_j(x + \hat{x}, y + \hat{y}) = p_j(x,y) + \hat{x}\,\frac{\partial p_j(x,y)}{\partial x}|_{x \to x+\hat{x}} + \hat{y}\,\frac{\partial p_j(x,y)}{\partial y}|_{y \to y+\hat{y}} + \cdots,$$

dividing through by h, and letting h tend to 0, yields

$$-b\,\frac{\partial p_1(x,y)}{\partial x} + a\,\frac{\partial p_1(x,y)}{\partial y} = -q_d\,p_1(x,y) + q_u\,p_0(x,y). \tag{2}$$

In State 0, we can analogously derive the partial differential equation

$$c\,\frac{\partial p_0(x,y)}{\partial x} = q_d\,p_1(x,y) - q_u\,p_0(x,y). \tag{3}$$

Similarly, by considering small line elements, the following ODEs, which couple the one- and two-dimensional probability densities, can be derived:

$$-d\,\frac{dw(y)}{dy} = -q_d\,w(y) + b\,p_1(0,y) \tag{4}$$

$$c\,\frac{ds(x)}{dx} = -q_u\,s(x). \tag{5}$$

By conservation of probability, the rate of flow into $(0,y)$ in State 0 must equal the rate of flow out of $(0,y)$ in State 0. Thus,

$$q_d\,w(y) = c\,p_0(0,y). \tag{6}$$

Similarly, since the rate of flow into $(x,0)$ in State 1 equals the rate of flow out of $(x,0)$ in State 1, it follows that

$$q_u\,s(x) = a\,p_1(x,0). \tag{7}$$

Also, since the rate of flow into the point $(0,0)$ in State 1 equals the rate of flow out of $(0,0)$ in State 1

$$d\,w(0) = q_d\,\gamma. \tag{8}$$

Finally, since the rate of flow into $(0,0)$ in State 0 equals the rate of flow out of $(0,0)$ in State 0, it follows that

$$q_d\,\gamma = c\,s(0). \tag{9}$$

Combining equations (2)-(9), and eliminating the one-dimensional densities $w(y)$ and $s(x)$, yields

$$-b\,\frac{\partial p_1(x,y)}{\partial x} + a\,\frac{\partial p_1(x,y)}{\partial y} = -q_d\,p_1(x,y) + q_u\,p_0(x,y), \tag{10}$$

$$c\frac{\partial p_0(x,y)}{\partial x} = q_d\, p_1(x,y) - q_u\, p_0(x,y), \tag{11}$$

$$-\frac{c\,d}{q_d}\frac{dp_0(0,y)}{dy} = -cp_0(0,y) + b\,p_1(0,y), \tag{12}$$

$$\frac{c}{q_u}\frac{dp_1(x,0)}{dx} = -p_1(x,0), \tag{13}$$

where

$$p_0(0,0) = \frac{q_d^2}{c\,d}\gamma \quad \text{and} \quad p_1(0,0) = \frac{q_d\,q_u}{a\,c}\gamma. \tag{14}$$

2 Steady-State Probability Densities

Solving equation (13), using (14), yields

$$p_1(x,0) = \frac{q_d\,q_u\,\gamma}{a\,c}\,e^{-\frac{q_u}{c}x}. \tag{15}$$

In order to determine the complete solution, we will use the Laplace Transform on the y-variable. For $j = 0, 1$, define

$$P_j(x,s) := \int_0^\infty p_j(x,y)\,e^{s\,y}\,dy.$$

It is straightforward to show the following transformations:

$$\frac{\partial p_j(x,y)}{\partial x} \longrightarrow \frac{d\,P_j(x,s)}{d\,x},$$

$$\frac{\partial p_j(x,y)}{\partial y} \longrightarrow s\,P_j(x,s) - p_j(x,0),$$

and

$$\frac{d\,p_j(0,y)}{d\,y} \longrightarrow s\,P_j(0,s) - p_j(0,0).$$

Applying these transformations to (10)-(12), using (15), yields the following set of ordinary differential equations:

$$-b\frac{d}{dx}P_1(x,s) + (a\,s + q_d)\,P_1(x,s) - q_u\,P_0(x,s) = \frac{q_d\,q_u\,\gamma}{c}e^{-\frac{q_u}{c}x} \tag{16}$$

$$c\frac{d}{dx}P_0(x,s) + q_u\,P_0(x,s) - q_d\,P_1(x,s) = 0 \tag{17}$$

$$c\left(\frac{d}{q_d}s - 1\right)P_0(0,s) + b\,P_1(0,s) = q_d\,\gamma. \tag{18}$$

Solving these equations yields

$$P_0(x,s) = A_0(s)\,e^{\alpha(s)x} - \frac{q_d\,\gamma}{c}\,e^{-\frac{q_u}{c}x} \quad \text{and} \quad P_1(x,s) = A_1(s)\,e^{\alpha(s)x},$$

where

$$A_0(s) = \frac{d\,q_d\,\gamma\,s}{b\,q_u - c\,q_d + c\,d\,s + b\,c\,\alpha(s)} \quad \text{and} \quad A_1(s) = \frac{d\,(q_u + c\,\alpha(s))\,\gamma\,s}{b\,q_u - c\,q_d + c\,d\,s + b\,c\,\alpha(s)}$$

with

$$\alpha(s) = \frac{a}{2\,b}(s + \beta - \sqrt{(s+\beta)^2 - \frac{4\,b\,q_d\,q_u}{a^2\,c}}) - \frac{q_u}{c}$$

and

$$\beta = \frac{b\,q_u + c\,q_d}{a\,c}.$$

Eliminating the square root from the denominator of $A_0(s)$ and $A_1(s)$ gives

$$A_0(s) = \frac{q_d\,\gamma}{c}\,[\frac{a\,(\theta - (s+\beta))}{2\,(a+d)\,(s+\lambda)} + \frac{q_d}{d\,(s+\lambda)} + 1] \tag{19}$$

and

$$A_1(s) = \frac{\gamma}{a+d}\,[\frac{a\,d}{2}\,(\frac{q_u}{c\,(a+d)\,(s+\lambda)} - \frac{1}{b})\,(\theta - (s+\beta)) + \frac{q_d\,q_u}{c\,(s+\lambda)}]. \tag{20}$$

where

$$\theta := \sqrt{(s+\beta)^2 - \frac{4\,b\,q_d\,q_u}{a^2\,c}},$$

and

$$\lambda := \frac{b\,q_u}{c\,(a+d)} - \frac{q_d}{d},$$

where $\lambda > 0$ is the capacity condition.

Since the resultant steady-state probability densities are continuous, we can move differentiation with respect to x outside of the integral. Thus, we have the following inverse Laplace transformations:

$$\frac{1}{\theta}\,Exp[-\frac{a\,x}{2\,b}(\theta - (s+\beta))] \longrightarrow e^{-\beta\,y}\,I_0(\epsilon\sqrt{y^2 + \frac{a\,x\,y}{b}}),$$

$$\frac{s+\beta}{\theta}\,Exp[-\frac{a\,x}{2\,b}(\theta - (s+\beta))] \longrightarrow e^{-\beta\,y}\,\frac{\partial}{\partial y}\,[I_0(\epsilon\sqrt{y^2 + \frac{a\,x\,y}{b}})] + \delta(y),$$

$$\frac{\theta - (s+\beta)}{\theta}\,Exp[-\frac{a\,x}{2\,b}(\theta - (s+\beta))] \longrightarrow -\frac{2\,b}{a}\,e^{-\beta\,y}\,\frac{\partial}{\partial x}\,[I_0(\epsilon\sqrt{y^2 + \frac{a\,x\,y}{b}})],$$

$$(\theta - (s+\beta))\,\frac{s+\beta}{\theta}\,Exp[-\frac{a\,x}{2\,b}(\theta - (s+\beta))] \longrightarrow -\frac{2\,b}{a}\,e^{-\beta\,y}\,\frac{\partial^2}{\partial x\,\partial y}\,[I_0(\epsilon\sqrt{y^2 + \frac{a\,x\,y}{b}})],$$

$$\frac{1}{\theta\,(s+\lambda)}\,Exp[-\frac{a\,x}{2\,b}(\theta - (s+\beta))] \longrightarrow e^{-\lambda\,y}\int_0^y e^{(\lambda-\beta)\,\tau}\,I_0(\epsilon\sqrt{\tau^2 + \frac{a\,x\,\tau}{b}})\,d\tau,$$

$$\frac{\theta - (s+\beta)}{\theta\,(s+\lambda)}\,Exp[-\frac{a\,x}{2\,b}(\theta - (s+\beta))] \longrightarrow -\frac{2\,b}{a}\,e^{-\lambda\,y}\,\frac{\partial}{\partial x}\,[\int_0^y e^{(\lambda-\beta)\,\tau}\,I_0(\epsilon\sqrt{\tau^2 + \frac{a\,x\,\tau}{b}})\,d\tau],$$

where $I_0(\cdot)$ is the modified Bessel function of order 0, i.e.,

$$I_0(x) = \sum_{k=0}^{\infty} \frac{(x/2)^{2k}}{(k!)^2},$$

and

$$\epsilon := \sqrt{\frac{4\,b\,q_d\,q_u}{a^2\,c}}.$$

Applying these transformations to equations (19) and (20), yields the steady-state probability densities in terms of the zero-order modified Bessel function:

$$\begin{aligned} p_0(x,y) &= e^{-(q_u/c\,x+\beta\,y)}\,[-\delta_1\,\frac{2\,b}{a}\,\frac{\partial}{\partial x}\,[I_0(\epsilon\sqrt{y^2+\frac{a\,x\,y}{b}})] - \\ &\quad \delta_2\,\frac{2\,b}{a}\,e^{-(\lambda-\beta)\,y}\,\frac{\partial}{\partial x}\,[\int_0^y e^{(\lambda-\beta)\,\tau}\,I_0(\epsilon\sqrt{\tau^2+\frac{a\,x\,\tau}{b}})\,d\tau] + \delta_3\,I_0(\epsilon\sqrt{y^2+\frac{a\,x\,y}{b}}) + \\ &\quad \delta_4\,\frac{\partial}{\partial y}\,[I_0(\epsilon\sqrt{y^2+\frac{a\,x\,y}{b}})] + \delta_5\,e^{-(\lambda-\beta)\,y}\,\int_0^y e^{(\lambda-\beta)\,\tau}\,I_0(\epsilon\sqrt{\tau^2+\frac{a\,x\,\tau}{b}})\,d\tau] \end{aligned}$$

and

$$\begin{aligned} p_1(x,y) &= e^{-(q_u/c\,x+\beta\,y)}\,[-\theta_1\,\frac{2\,b}{a}\,\frac{\partial}{\partial x}\,[I_0(\epsilon\sqrt{y^2+\frac{a\,x\,y}{b}})] - \theta_2\,\frac{2\,b}{a}\,\frac{\partial^2}{\partial x\,\partial y}\,[I_0(\epsilon\sqrt{y^2+\frac{a\,x\,y}{b}})] - \\ &\quad \theta_3\,\frac{2\,b}{a}\,e^{-(\lambda-\beta)\,y}\,\frac{\partial}{\partial x}\,[\int_0^y e^{(\lambda-\beta)\,\tau}\,I_0(\epsilon\sqrt{\tau^2+\frac{a\,x\,\tau}{b}})\,d\tau] + \theta_4\,I_0(\epsilon\sqrt{y^2+\frac{a\,x\,y}{b}}) + \\ &\quad \theta_5\,e^{-(\lambda-\beta)\,y}\,\int_0^y e^{(\lambda-\beta)\,\tau}\,I_0(\epsilon\sqrt{\tau^2+\frac{a\,x\,\tau}{b}})\,d\tau], \end{aligned}$$

where

$$\delta_1 = \frac{(a+2d)\,q_d\,\gamma}{2\,c\,(a+d)},\ \delta_2 = \frac{d\,q_d\,\gamma}{2\,c(a+d)}[\frac{(a+d)\,q_d}{d^2} - \frac{b\,q_u}{c\,(a+d)}],\ \delta_3 = \frac{q_d^2\,\gamma}{c\,d},$$

$$\delta_4 = \frac{q_d\,\gamma}{c},\ \delta_5 = \frac{q_d^2\,\gamma}{a\,c}\,[\frac{(a+d)\,q_d}{d^2} - \frac{b\,q_u}{c\,(a+d)}],$$

and

$$\theta_1 = -\frac{a\,d\,q_u\,\gamma}{2\,c\,(a+d)^2},\ \theta_2 = \frac{a\,d\,\gamma}{2\,b\,(a+d)},\ \theta_3 = \frac{d^2\,q_u\,\gamma}{2\,c\,(a+d)^2}\,[\frac{(a+d)\,q_d}{d^2} - \frac{b\,q_u}{c\,(a+d)}],$$

$$\theta_4 = \frac{(a+2\,d)\,q_d\,q_u\,\gamma}{a\,c\,(a+d)},\ \theta_5 = \frac{d\,q_d\,q_u\,\gamma}{a\,c\,(a+d)}[\frac{(a+d)\,q_d}{d^2} - \frac{b\,q_u}{c\,(a+d)}].$$

Thus, the steady-state probability density functions are known for any system which can be transformed so that it has flow diagrams of the form shown in Figure 1. It follows that, for any such system, it is straightforward to evaluate any cost function which depends only on the steady-state buffer level probability densities. In the following section, we will discuss an application of this result.

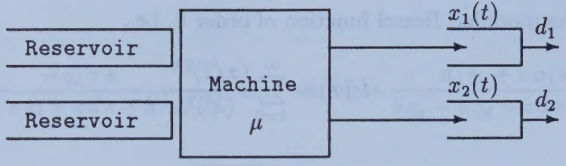


Figure 2: The single machine, two part-type system.

3 One Machine, Two Part-Types

Consider the single machine, two part-type system shown in Figure 2. Let $x_j(t)$ be the level of buffer b_j at time t. Thus, $x_j(t) \in [-\infty, \infty]$, where a negative buffer level indicates a shortfall. Buffer b_j is continuously depleted at a constant rate d_j, indicating a demand for part-type j. Assume that there is an infinite surplus of raw material for each part-type, which is allowed to flow into the system at rate $u_j(t)$, where $u_1(t) + u_2(t) \leq \mu$. The machine occasionally fails. The status process of the machine is modeled as a continuous time Markov chain, with q_u being the rate of transition from down to up and q_d being the rate of transition from up to down. The machine cannot process any material when it is in the down state.

Suppose that a linear switching curve control is implemented by the machine. That is, when the machine is up, the control attempts to regulate the buffer levels so that they lie on the line defined by

$$(\mu_2 - d_2)(x_1 - z_1) = (\mu_1 - d_1)(x_2 - z_2),$$

and they do not exceed the hedging point (z_1, z_2). Specifically, when the machine is up,

$$(u_1, u_2) = \begin{cases} (0,0) & \text{if } x_1 > z_1, x_2 > z_2 \\ (d_1, d_2) & \text{if } x_1 = z_1, x_2 = z_2 \\ (\mu, 0) & \text{if } (\mu_2 - d_2)(x_1 - z_1) < (\mu_1 - d_1)(x_2 - d_2) \\ (\mu_1, \mu_2) & \text{if } (\mu_2 - d_2)(x_1 - z_1) = (\mu_1 - d_1)(x_2 - d_2) \\ (0, \mu) & \text{if } (\mu_2 - d_2)(x_1 - z_1) > (\mu_1 - d_1). \end{cases}$$

The linear switching curve control is parameterized by the choice of μ_1, with $\mu_2 := \mu - \mu_1$.

In the case that $\mu_1/d_1 < \mu_2/d_2$, the direction and magnitude of material flow in the non-transient region (in $x_1 \times x_2$ space) is shown in Figure 3. The steady-state probability densities can be computed explicitly using the earlier result by applying the transformation

$$x := -(\mu_2 - d_2)(x_1 - z_1) + (\mu_1 - d_1)(x_2 - d_2),\ y := d_2\,(x_1 - z_1) - d_1\,(x_2 - z_2).$$

Thus, it is straightforward to compute any cost which is a function of the average buffer levels. Then, μ_1 can be chosen to minimize this cost function.

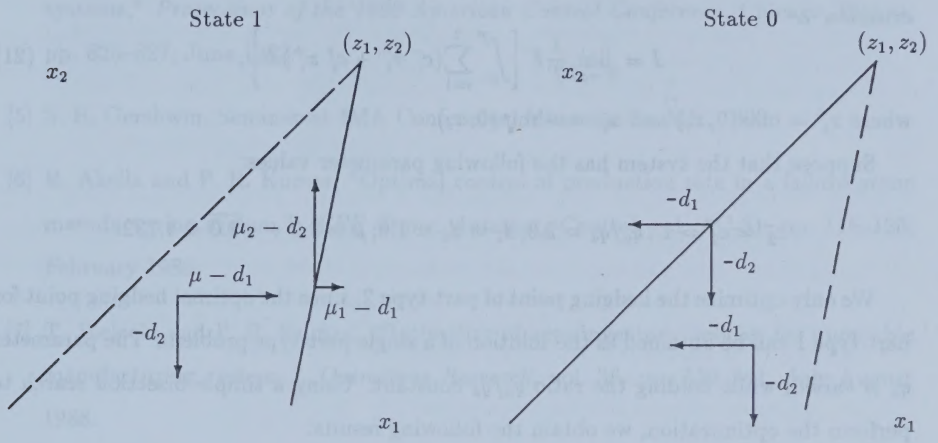


Figure 3: Flows and densities for the single machine, two part-type case.

There are several reasons for choosing a control of this form. First, if we set $\mu_1 = \mu$, then the resulting control is a prioritized hedging point (PHP) policy.[1] The PHP policy is optimal for the single part-type average and discounted cost problems (see [7] and [6]), even when there are multiple machine states (see [8] and [9]). Also, it is common in the literature to investigate this form of control for more complicated problems (see, for example, [4] and [11]).

In addition, for the average cost problem, the linear switching curve control includes the quadratic approximation to the cost function, which has been investigated in some detail (see, for example, [3], [1], and [2]).

Another reason for using the linear switching curve control is that, for the average additive cost criterion, it will yield the Just-In-Time (JIT) control whenever JIT is optimal (see [10] and [5]).

[1]That is, part-type 1 has priority over part-type 2, and the machine will feed buffer b_1 at rate μ as long as $x_1 < z_1$.

4 Numerical Examples

Using the steady-state probability densities derived in Section 2, we can numerically compute the optimal hedging points under a PHP policy for the average additive cost criterion

$$J = \lim_{T\to\infty} \frac{1}{T} E\left[\int_0^T \sum_{i=1}^{2} (c_i^- x_i^- + c_i^+ x_i^+) dt\right], \tag{21}$$

where $x_i^+ = \max(0, x_i)$ and $x_i^- = -\min(0, x_i)$.

Suppose that the system has the following parameter values:

$$c_2^- = c_2^+ = 1.; q_u/q_d = 2.0; d_1 = d_2 = 1.0; \mu = 3.0 + \sqrt{3.0} = 4.732.$$

We only optimize the hedging point of part-type 2, since the optimal hedging point for part-type 1 can be obtained as the solution of a single part-type problem. The parameter q_d is varied, while holding the ratio q_u/q_d constant. Using a simple bisection search to perform the optimization, we obtain the following results:

q_d	z_2
0.25	11.48
0.5	7.81
1.0	4.55
2.0	2.87
3.0	1.58
4.0	1.19

It should be noted that z_2, the optimal hedging point for part-type 2, decreases as q_d increases. This is intuitively clear, since an increase in q_d results in increasing q_u by twice as much, thus increasing the available capacity, i.e.,

$$(1/q_d)(\mu - d_1 - d_2) - (1/q_u)(d_1 + d_2).$$

References

[1] E. K. Boukas and A. Haurie, "Manufacturing Flow Control and Preventive maintenance: A Stochastic Control Approach," *IEEE Trans. Automatic Control*, vol. 35, pp. 1024–1031, 1990.

[2] M. Caramanis and G. Liberopoulos, " Perturbation Analysis of Flexible Manufacturing System Flow Controllers," *Operations Research*, vol. 40, pp. 1107–1125, November–December 1992.

[3] S. Gershwin, R. Akella and Y. Choong, "Short term production scheduling of an automated manufacturing facility," *IBM J. Res. Dev.*, vol. 29, pp. 392–400, 1985.

[4] J. R. Perkins and P. R. Kumar, "Buffer level control of unreliable manufacturing systems," *Proceedings of the 1992 American Control Conference*, Chicago, Illinois, pp. 826–827, June 1992.

[5] S. B. Gershwin, Seminar at IMA Conference, Minneapolis, MN, 1993.

[6] R. Akella and P. R. Kumar, "Optimal control of production rate in a failure prone manufacturing system," *IEEE Trans. Automat. Control*, vol. AC-31, pp. 116–126, February 1986.

[7] T. Bielecki and P. R. Kumar, "Optimality of zero-inventory policies for unreliable manufacturing systems," *Operations Research*, vol. 36, pp. 532–541, July-August 1988.

[8] A. Sharifnia, "Production control of a manufacturing system with multiple machine states," *IEEE Trans. Automat. Control*, vol. AC-33, pp. 600–626, July 1988.

[9] P. Glasserman, "Hedging-point production control with multiple failure modes," tech. rep., Columbia University, New York, N.Y., December 1993.

[10] N. Srivatsan, "Synthesis of optimal policies for stochastic manufacturing systems, " Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 1993. Also, Laboratory for Manufacturing and Productivity Rept. LMP-93-009, MIT.

[11] A. Cornelius, "An iterative simulation method for determining the optimal buffer level regulation policy for unreliable manufacturing systems," M.S. thesis, University of Illinois, Urbana, IL, 1992.

A CLEARING ROUND-ROBIN-BASED STABILIZATION MECHANISM

ANA FLORA P. DE C. HUMES
Depto. de Matematica Aplicada
Instituto de Matematica e Estatistica
Universidade de S. Paulo - Brazil
anaflora@ime.usp.br

CARLOS HUMES, JR
Depto. de Ciencia de Computacao
Instituto de Matematica e Estatistica
Universidade de S. Paulo - Brazil
humes@ime.usp.br
humes@riccati.csl.uiuc.edu

Abstract

This paper is devoted to stability problems within the Perkins-Kumar framework [6] for distributed scheduling of manufacturing systems.

The first result presented here is the existence of an unique limit cycle for a class of open-loop clearing policies (CGRR), allowing for self-loops in the considered machine. The previous results ([4],[5]) did not consider self-loops.

Using this result, a new stabilization mechanism for general multi-machine manufacturing systems (MGRR) is presented.

1 The Setting

1.1 General Setting

We shall consider manufacturing systems modelled according to the framework laid in [6], although with slightly different notation. Thus, we have a finite set $\mathcal{M} = \{1, 2, \ldots, M\}$ of machines and a finite set $\mathcal{P} = \{1, 2, \ldots, P\}$ of products. Each product p demands n_p tasks to be performed in the system, and let $\mathcal{I} = \{1, 2, \ldots, I\}$ be the set of such tasks, where $I = \sum_{p=1}^{P} n_p$. Each task i will be assigned to exactly one machine (denoted $m(i)$) and will service just one product (denoted $p(i)$).

Moreover, we will assume that each product p has a pre-specified order $\mathcal{P}_p$ in which its associated tasks have to be performed. The ordered n_p tuple that describes such order is called the *product stream* of p. This allow us to refer to the first product in the product stream $I(p)$ and for each task i to define its predecessor (i_-) in the product stream. I.e., if $\mathcal{P}_p = (i_p^1, i_p^2, \ldots, i_p^{n_p})$ is the sequence of tasks (product stream) associated with product p, if $i = i_p^{(k+1)}$ then $i_- = i_p^k$ and we will denote $(i_p^1)_- = I(p)_- = \emptyset$.

A task is in *self loop* if its input is the output of a task allocated to the same machine ($m(i) = m(i_-)$).

The machines are assumed to be indivisible, i.e., while machine m is processing a task, m is completely devoted to it. Each task i is serviced at a maximal rate τ_i^{-1}, when

*The research reported here has been partially supported by CNPq, IBM-Brasil, FAPESP, NSF-ECS-94403571.

its buffer is not empty and if a machine switches from servicing task i to servicing task i', a set-up time $\delta_{ii'}$ occurs. (Clearly $\delta_{ii'} \geqq 0$.)

This indivisibility may lead to the existence of parts waiting processing at the machines. We shall assume that each task i has an associated buffer b_i[1], whose contents at time t, will be denoted by $x_i(t)$. Denoting $e_p(t)$ the cumulative input (in the system) of product p in the interval $[0,t]$, $u_i(.)$ the cumulative input for task i, $y_i(t)$ the cumulative output of task i and using the convention $u_{I(p)}(t) = e_p(t) = y_{I(p)-}(t)$, we have

$$x_i(t) = x_i(0) + u_i(t) - y_i(t) = x_i(0) + y_{i-}(t) - y_i(t). \tag{1}$$

Clearly, the behavior of the system will depend upon the inputs to which it is submitted and to its operating rules (scheduling policy). The two most studied class of inputs are *linear deterministic* (LD) ($e_p(t) = d_p t,\ p \in \mathcal{P}$) and bounded burstiness (BB) with parameters ($(d_p, \gamma_p)\ p \in \mathcal{P}$), where

Definition: A function $f(.)$ said to be *bounded burstiness* (BB) with parameters (r, η) if

$$\forall t \geqq s \geqq 0,\quad f(t) - f(s) \leqq r(t-s) + \eta. \quad \square$$

Every LD input is also a BB input and the interest in studying LD inputs can be almost considered as a preparation to the analysis of the behavior of systems under BB inputs.

The scheduling policies we shall concentrate upon are decentralized, i.e., each machine decides upon its scheduling based on information of its state. An important property we wish to guarantee is that the number of parts waiting service remains bounded. This is a rough statement of idea of scheduling policy being stable. Clearly, stability will depend upon the class of allowed inputs we are considering, but we will drop this qualification (considering it implicitly done) unless necessary for the clarity of the text.

Definition: A policy is said to be *stable* for a system if given any initial buffer contents $\{x_p(0)\}_{i \in \mathcal{I}}$, there exist L such that

$$\forall t \geqq 0,\ \forall i \in \mathcal{I},\quad \|x_i(t)\| \leqq L. \quad \square$$

It can be shown [6], that stability is also equivalent to the existence of M (dependent on the initial conditions), such that

$$\forall t \geqq 0,\ \ \forall i \in \mathcal{I} \quad \|y_i(t) - e_{p(i)}(t)\| \leqq M.$$

It is quite natural that for the study of stable policies, we shall confine ourselves to policies that are stable for machines taken in isolation, i.e., systems formed by a single machine. Taking this fact into account it is clear that an important parameter is the congestion level (average load) at each machine defined as:

[1]For simplicity, we shall refer to b_i as buffer i.

$$\bar{\rho}_m := \sum_{i:m(i)=m} d_{p(i)}\tau_i, \qquad m = 1, 2, \ldots, M.$$

It will be convenient to introduce the congestion level of task i, that is defined as

$$\rho_i = d_{p(i)}\tau_i \qquad i = 1, 2, \ldots, I.$$

A natural condition when studying stability is *the capacity condition* (CC):

$$\forall m = 1, 2, \ldots, M, \qquad \bar{\rho}_m := \sum_{i:m(i)=m} \rho_i < 1 \tag{2}$$

It must be noted that in the absence of set-up times, the indivisibility of machines would be meaningless, unless we assume a strictly positive minimum for the duration of each production run. Also, if all set-up times are zero, any machine in isolation, with BB inputs obeying (CC) and operating under a work conserving policy (i.e., any policy where the machine will work on some task, at its maximal rate, as long some buffer of machine is non-empty), will be stable.

2 Periodicity of one machine under CGRR policies

In this section we will concentrate in the study of a single machine ($M = 1$), subject to LD inputs. We shall do so in order to prove that there is a class of scheduling policies that has a limit cycle, i.e. , the trajectory of the system converges asymptotically to a periodic function. In order to do so let us define:

Definition: Eventually Periodic Policy

A scheduling policy is *eventually periodic* if for any machine subject to LD inputs (obeying CC), for any initial state $x(0)$, there are $(T_0, N) \in \mathbf{R}_+ \times \mathbf{N}$ and $i : \{0, 1, \ldots N\} \to \mathcal{I}$, such that:

(i) There is partition $\{I_l\}_{l \in \mathbf{N}}$ of $[T_0, \infty)$, where $I_l = [a_l, b_l)$ and $a_{l+1} = b_l$;

(ii) $i(N) = i(0)$ and $\forall t \in I_l$, at time t , the machine services task $i(l \bmod N)$.

Under these conditions, we shall refer to $(N, i(.))$ as the *eventually periodic behavior*, where N is its *eventual period* and $i(.)$ its *indicator function.* □

As every natural number l can be written as [(k-1)N+j], with $1 \leq j \leq N$, we shall refer to the intervals I_l , as in (ii) above as the j^{th} *time slice of the* k^{th} *(eventual) cycle* and we shall denote its *duration*, i.e. $(b_l - a_l)$, by t_j^k. Note that the j^{th} time slice of any cycle services the same product $(i(j))$. We shall only study cases where the the eventually periodic behavior is such that $i(\{1, 2, \ldots N\}) = \mathcal{I}$.

Definition: Stable policies with limit cycle

A scheduling policy is *stable with limit cycle* (SLC) if for any machine subject to LD inputs (obeying CC), it is stable, eventually periodic and, for any initial state $x(0)$ and

for all j, there is $\bar{t}_j$ such that $\{t_j^k\}_{k\in \mathbf{N}} \to \bar{t}_j$. Under these conditions, we shall refer to $\bar{t}$ as the *limit cycle* of the policy. □

A natural way to try to obtain a SLC policy is to impose an eventually periodic behavior and try to define the duration of the time slices in a convenient form. One form to define the time slices' duration would be to impose that the policy is clearing:

Definition: Clearing Policy
A scheduling policy is *clearing* if *all* of its production runs are clearing, i.e., if a task i is being processed at $m(i)$ this processing will continue till the buffer of such task is empty (and some other buffer $m(i)$ is non-empty). □

A fact [6] that we shall make usage further on, is that the duration Δ of a clearing run for task i, starting at time $\hat{T}$, fed during that run, by a LD input of rate d, will be given by:

$$\Delta = (1 - d\tau_i)^{-1}[\tau_i x_i(\hat{T}) + \delta]. \tag{3}$$

The class of policies we shall concentrate on is:

Definition: Generalized Clearing Round-Robin Policy (CGRR)
A *Clearing Generalized Round-Robin* policy is one that is clearing and to which an eventual periodic behavior $(N, i(.))$ is imposed. □

Due to the clearing nature of CGRR, in order to prove its stability, it suffices to show that the production runs are bounded, so to prove that CGRR is a SLC policy, it suffices to show the convergence of $\{t_j^k\}_{k\in \mathbf{N}}$.

In order to simplify notation, let us denote by t and t' the N-dimensional vectors defined by $t_j = t_j^{k+1}$ and $t'_j = t_j^k$, $j = 1, 2, \ldots, N$. Moreover, due to the eventually periodic behavior, the set-up in the j^{th} production run of any cycle is given by $\delta_j = \delta_{i(j-1)i(j)}$.

Let us now consider a task i and J_i the set of indexes of production runs devoted to processing it during an eventual period, i.e.,

$$J_i = \{j = 1, 2, \ldots, N \ / \ i(j) = i\} = \{\alpha_1, \alpha_2, \ldots, \alpha_S\},$$

where, without loss of generality, $\alpha_1 < \alpha_2 < \ldots < \alpha_S$. We should have denoted $J_i = \{\alpha_1^i, \alpha_2^i, \ldots, \alpha_{S_i}^i\}$, but, for simplicity, we use the notation without explicit task indication.

Then, if $i_- = \emptyset$, as in [5], (3) becomes:

$$t_{\alpha_1} = \delta_{\alpha_1} + \rho_i\Big(\sum_{l=\alpha_S+1}^{N} t'_l + \sum_{l=1}^{\alpha_1} t_l\Big), \tag{4}$$

$$t_{\alpha_{(s+1)}} = \delta_{\alpha_{(s+1)}} + \rho_i \sum_{l=\alpha_s+1}^{\alpha_{(s+1)}} t_l, \qquad \text{for } s = 1, \ldots, S-1. \tag{5}$$

Moreover, if $i_- \neq \emptyset$, analogously (3) becomes:

$$t_{\alpha_1} = \delta_{\alpha_1} + (\rho_i/\rho_{i_-})\Big(\sum_{\substack{l=\alpha_S+1 \\ i(l)=i_-}}^{N} (t'_l - \delta_l) + \sum_{\substack{l=1 \\ i(l)=i_-}}^{\alpha_1-1} (t_l - \delta_l)\Big) \tag{6}$$

$$t_{\alpha_{(s+1)}} = \delta_{\alpha_{(s+1)}} + (\rho_i/\rho_{i_-})\Big(\sum_{\substack{l=\alpha_s+1 \\ i(l)=i_-}}^{\alpha_{(s+1)}-1} (t_l - \delta_l)\Big), \qquad \text{for } s = 1, \ldots, S-1. \tag{7}$$

Considering equations (4),(5),(6) and (7), for all tasks i, we obtain a linear system:

$$t' = \bar{B}t' + \bar{C}t + \bar{\delta}, \tag{8}$$

where $\bar{B}$ and $\bar{C}$ have non negative entries.

A sufficient condition to guarantee not only the stability of CGRR, but also the existence of an unique limit cycle is that all the eigenvalues of $[(I - \bar{B})^{-1}\bar{C}]$ lye in the interior of the unity circle. We shall show that, under (CC), this is the case.

Before exploiting the particularities of the linear system (8), let us review some results for square matrices with non negatives entries. We shall use, for $N \times N$ matrices, the following notation :

$F \geqq G \Leftrightarrow f_{ij} \geqq g_{ij}, \quad \forall i, j = 1, 2, \ldots, N;$
$\mathcal{G}_N = \{M \in \mathbf{R}^{N\times N} \,/\, M \geqq 0\} = \mathbf{R}_+^{N\times N};$
$\bar{\lambda}(F) =$ spectral radius of F.

The first two of the following lemmas are classical, so no proof will be presented. For the third one, due to Thomas I. Seidman, just a bare sketch of its demonstration is given.

Lemma 1. *If $M \in \mathcal{G}_N$, then*

(a) $\lambda = \bar{\lambda}(M)$ is an eigenvalue of M, with an eigenvector in $\mathbf{R}_+^N$;

(b) $\forall \alpha \geqq 0$, $\bar{\lambda}(\alpha M) = \alpha\bar{\lambda}(M)$;

(c) $\bar{\lambda}(M) < 1 \Rightarrow (I - M)$ is non singular and $(I - M)^{-1} \geqq 0$. □

Lemma 2. *The function $\bar{\lambda} : \mathcal{G}_N \to \Re_+$ is nondecreasing, i.e.,*

$$F \geqq G \geqq 0 \Rightarrow \bar{\lambda}(F) \geqq \bar{\lambda}(G) \geqq 0. \quad \square$$

These two lemmas are preparatory to the next one:

Lemma 3 (Seidman) [7]. *Let $(B, C) \in \mathcal{G}_N \times \mathcal{G}_N$ such that $\bar{\lambda}(B + C) \leqq \alpha$. Then:*

$$\bar{\lambda}((I - B)^{-1}C) \leqq \alpha.$$

Proof. The proof can be done by contradiction, using the previous lemmas: Let $A = ((I - B)^{-1}C)$ and assume $\lambda = \bar{\lambda}(A) \leqq 1$ (else use rescaling), with $\lambda x = Ax$, so x is an eigenvector of $(\lambda B + C)$ so $\lambda \leqq \bar{\lambda}(\lambda B + C) \leqq \alpha$. □

Using these results, we can state:

Theorem 1. *CGRR is a SLC policy.*

Proof. Lemmas 1, 2 and 3 imply that a sufficient condition for the result to hold is that $\bar{\lambda}(\bar{B} + \bar{C}) < 1$, for $\bar{B}$ and $\bar{C}$ in (8).

If $\lambda = \bar{\lambda}(\bar{B} + \bar{C})$, then there is a $v \in \mathbf{R}_+^N$ ($v \neq 0$), such that :

$$i(j) = i, i_- = \emptyset \quad \Rightarrow \lambda v_{\alpha_{(s+1)}} = \rho_i \sum_{l=\alpha_s+1}^{\alpha_{(s+1)}} v_l \qquad s = 0, 1, \ldots, (S-1), \tag{9}$$

$$i(j) = i, i_- \neq \emptyset \quad \Rightarrow \lambda v_{\alpha_{(s+1)}} = (\rho_i/\rho_{i_-}) \sum_{\substack{l=\alpha_s+1 \\ i(l)=i_-}}^{\alpha_{(s+1)}} v_l \qquad s = 0, 1, \ldots, (S-1), \tag{10}$$

with the convention $\sum_{\alpha_0+1}^{\alpha_1} v_l = \sum_{\alpha_s+1}^{N} v_l + \sum_{l=1}^{\alpha_1} v_l$.

Defining $T_i = \sum_{i(j)=i} v_j$ and adding up equations (9) and (10) for all production runs dedicated to the same task, we verify that $\{T_i\}_{i=1}^I$ are such that:

$$\begin{cases} a1)\ \exists \tilde{i} \ :\ T_{\tilde{i}} \neq 0 & \\ a2)\ i_- = \emptyset & \Rightarrow \quad \lambda T_i = \rho_i \sum_{i'=1}^I T_{i'} \\ a3)\ i_- \neq \emptyset & \Rightarrow \quad \lambda T_i = (\rho_i/\rho_{i_-}) T_{i_-} \end{cases}$$

Denoting $L = \max\{n_p \ / \ p \in \mathcal{P}\}$ and $c = \rho^{1/L}$ $(c < 1)$, let us scale T_i as

$$s_i = \rho_i^{-1} c^{\mu_i} T_i, \quad i = 1, \ldots, I$$

where $\mu_i + 1 =$ position of task i in its product stream. Then:

$$\lambda s_i = \begin{cases} \sum_{i'=1}^I c^{-\mu_{i'}} \rho_{i'} s_{i'} & \text{if } i_- = \emptyset \ \ (\mu_i = 0) \\ c s_{i_-} & \text{if } i_- \neq \emptyset \ \ (\mu_i \geqq 1) \end{cases}$$

Let $s_{\tilde{i}} = \|s\|_\infty$. As $\{T_i\}_{i=1}^I$ are not all zero, $\|s\|_\infty > 0$.

If $\tilde{i}_- = K \neq \emptyset$, then $\qquad \lambda = c \frac{s_K}{\|s\|_\infty} \leqq c.$

If $\tilde{i}_- = \emptyset$, then

$$\begin{aligned} \lambda = \ & \textstyle\sum_{i'=1}^I c^{-\mu_{i'}} \rho_{i'} \frac{s_{i'}}{\|s\|_\infty} \leqq \\ & \leqq \textstyle\sum_{i'=1}^I c^{-\mu_{i'}} \rho_{i'} \leqq c^{-(L-1)} \sum_{i=1}^I \rho_i = \\ & = (\rho^{1/L})^{-(L-1)} \rho \ = \ \rho^{1/L} \ = \ c. \end{aligned}$$

So, $\bar{\lambda}[(I - \bar{B})^{-1}\bar{C}] \leqq \bar{\lambda}(\bar{B} + \bar{C}) = \lambda \leqq c = \rho^{1/L} < 1$,
and the result follows with $\bar{t} = [(I - \bar{B} - \bar{C})]^{-1}\bar{\delta}$. □

An important consistency property of the above result is that $\bar{t}_j$ must be non inferior to δ_j, for $j = 1, \ldots, N$. There is a trivial consequence of theorem 1, as given any initial

condition, it is trivial to verify that $(t_j^k \geqq \delta_j)_{j=1}^J$ for all k. So the fact that the sequence $\{t^k\}$ converges to $\bar{t}$, implies $\bar{t}_j \geqq \delta_j$, $j = 1, \ldots, N$. If we take into account that equations (4) to (7) have to be obeyed with $t'_j = t_j = \bar{t}_j$, it is an easy exercise to verify that if there is at least one non zero set up time, then $(\bar{t}_j > \delta_j,\ j = 1, \ldots, N)$, otherwise[2] $\bar{t}_j = 0$.

3 Modified CGRR: a stabilization technique.

An interesting application of the above results is that we can devise a protocol that induces stability for any system with M machines, provided that (CC) is obeyed, independently of the system being or not acyclic.

Suppose for each machine m, in the system, we calculate the limit cycle of each machine operating under a CGRR with linear deterministic inputs. As each machine may have different eventual behaviors $(N_m, i_m(.))$ call each policy $\mathrm{GRR_m}$, with eventual period N_m. Call the duration of each such time slice $\bar{t}_j^m$, i.e., the duration of the j^{th} time slice of the limit cycle of machine m.

Given these numbers, operate each machine with its clearing open loop policy $\mathrm{CGRR_m}$ with provision that no j^{th} time slice can exceed $\bar{t}_j^m$. Call this policy $\mathrm{MGRR_m}$, i.e., *modified CGRR for machine m*. We shall say that a system operates under MGRR if each machine m of the system is operating under a policy of the family $\mathrm{MGRR_m}$.

An important result is that this policy not only is stable, but we can also derive bounds for the maximum content of each buffer. As these ideas are a simple variation of bounding techniques used elsewhere ([1], [5]), we shall prove stability of the policy using a very naive bounding technique.

Theorem 2. *Any system operating under modified generalized round-robin is stable for bounded burstiness inputs that obey (CC).*

Proof. Along the proof, we shall refer to time slices being *full* if their duration is equal to the upper limit imposed by MGRR (otherwise they will be called *partial*). I.e., the time slice j, in the k^{th} N_m cycle, for machine m, is said to be full if and only if

$$(t_j^k = \bar{t}_j^m \text{ and } m(i(j)) = m).$$

The key point of the proof is that in the limit cycle for a stable policy under LD inputs, the amount of parts entering a buffer is exactly equal to what it processes. Clearly, this implies the stability of the buffers associated with self loops.

Using this fact, we will prove stability by induction along the the product streams, considering only buffers that are not self loops.

[2] Remember that $\delta \geqq 0$.

Consider then $i \in \mathcal{P}_p$ and let us assume that $u_i(.)$ is such that

$$\forall t \geqq s \geqq 0, \qquad u_i(t) - u_i(s) \leqq d_p(t-s) + \gamma_i.$$

At least one such i exists, nominally $i = I(p)$ and $\gamma_i = \gamma_p$. (clearly, this is the basis of the induction).

We shall partition the real line in two class of intervals: an interval will be called full (partial) if all production runs of task i in that interval are full (partial). The end of such interval will be the end of the last production run that characterizes the interval. Except for the initial interval, any full (partial) interval begins at the end of the closest partial (full) run for product p at machine $\bar{m} = m(i)$.

Due to the clearing nature of the policy, it is clear that for any partial interval PI,

$$t \in PI \quad \Rightarrow \quad x_i(t) \leqq \max_{j:i(j)=i} [(1-\rho_i)\tau_i^{-1}(\bar{t}_j^{\bar{m}} - \delta_j)].$$

Let us consider now a full interval FI starting at time T_0, by processing a time slice indexed $j(T_0)$. If $T_0 = 0$, $x_i(T_0) = x_i(0)$ and $j(T_0) = 1$. Otherwise, at T_0, a partial time slice indexed j^* $(i(j^*) = i)$ has been completed so $x_i(T_0) = 0$ and $j(T_0) = [(j^* \bmod \mathrm{N}_{\bar{m}}) + 1]$. Now, for every t in FI, let us define

$T_t = \max\{t' \in [T_0, t]$ such that at t' a time slice indexed $j(T_0)$ has been initiated$\}$.

From the definition of T_t it follows that

$$(t - T_t) < T^{\bar{m}} = \sum_{j=1}^{N_{\bar{m}}} \bar{t}_j^{\bar{m}}$$

and

$$y_i(t) - y_i(T_0) \geqq y_i(T_t) - y_i(T_0) \geqq d_p(T_t - T_0).$$

This last inequality indicates that between T_0 and T_t all time slices of all tasks in machine $\bar{m}$ were full, the output of the full time slices of task i should be equal to the output in the cycle for LD inputs and this is equal to $d_p(T_t - T_0)$. If the time slices of other tasks were not full, the accumulated output of task i will not decrease (it may even increase if more time slices were dedicated to task i then what would be in the LD cycle).

So

$$\begin{aligned} x_i(t) &= x_i(T_0) + u_i(t) - u_i(T_0) - (y_i(t) - y_i(T_0)) = \\ &\leqq x_i(0) + d_p(t - T_0) + \gamma_i - (d_p(T_t - T_0)) = \\ &= x_i(0) + d_p(t - T_t) + \gamma_i \\ &\leqq x_i(0) + d_p T^{\bar{m}} + \gamma_i \end{aligned}$$

The conclusion of the proof is done by induction in each product path, remembering that the output of a bounded buffer subject to a BB input is a BB function, with the same rate as the input. □

It is important to note that the above stabilizing mechanism is well defined if and only if $\bar{t}_m > 0$. This is the case when at least one set up time is not zero in each machine. Obviously, if there are no set up times a variation of this policy can be implemented by the introduction of an artificial set up time (some slack). This comment goes along with a line of discussion in the area that for certain policies the introduction of delays and or back off may be convenient in order to stabilize the system.

Moreover, a careful reading of the above demonstration will show that we assumed that every time slice occurs in MGRR, i.e., if at time $\hat{T}$ a time slice $\bar{j}$ associated with task $\bar{i}$ is to occur according to MGRR, it will occur if $x_{\bar{i}}(\hat{T}) = 0$ or, even worse, $x_{\bar{i}}(t) = 0$ for $t \in [\hat{T}, \hat{T} + \delta_{\bar{j}}]$. This implies in a certain inefficiency similar to the one in CAF with back off [6], in the sense we allocate production time for no use at all. This assumption is important as MGRR makes use only of the knowledge of values of set up times in the sequence they occur, i.e., $\delta_{i(j)i(j+1)}$. This "backing off" is not necessary if some "triangle inequality" holds (like the quite natural $\delta_{j^1 j^3} \leqq \delta_{j^1 j^2} + \delta_{j^2 j^3}$).

4 Perspectives and comparison with previous work

The main contribution of this paper is to present a new stabilizing protocol: the modified clearing generalized round-robin policy (MGRR). In order to do so we had to exploit the ideas of eventual periodicity and we made use of existence of a limit cycle for the clearing generalized round robin policies (CGRR).

The first contribution of this work was to show the existence of a limit cycle for CGRR policies, allowing for the existence of self loops (not covered in previous work). This was necessary in order to be able to present MGRR as a general stabilizing mechanism.

This stabilization method although similar to CAF with back off [6] and also to regulators [2] is conceptually different and resembles the implementation of the USSM [3]. There are at least three main differences with previous work, besides technicalities (like the proof here presented does not use any Lyapunov function):

(i) In USSM, we have a service discipline to the priority queue that is a modified FCFS;

(ii) In CAF with back off, we do have a policy that is not work conserving;

(iii) Finally, if we try to emulate the MGRR policy with the usage of regulators, we will go out of the strict interpretation of such devices. To see this, it suffices to note that if task 1 has a large load (multiple of $\bar{t}_1^m$) and the other buffers are empty, then this buffer will run at speed τ_{p1}^{-1} for some period and not at d'_p, such that $d'_p \geq d_p$ and $\sum \tau_{pi} d'_p < 1$.

Due to this reasoning, we claim that this is apparently a regulating mechanism different in nature of all the previous works. We believe that this is simply a variation of round robin with packets, whose stability is the subject of a forthcoming work.

References

[1] R. L. Cruz. A calculus for network delay, part II: Network analysis. *IEEE Transactions on Information Theory*, 37(1):132–141, January 1991.

[2] C. Humes, Jr. A regulator stabilization technique: Kumar-Seidman revisited. To appear in *IEEE Transactions on Automatic Control*, 1993.

[3] P. R. Kumar and T. I. Seidman. Dynamic instabilities and stabilization methods in distributed real-time scheduling of manufacturing systems. AC-35(3):289–298, March 1990.

[4] J. R. Perkins. Stable, real-time scheduling of a single machine manufacturing systems. Master's thesis, University of Illinois, Urbana, IL, 1990. Also, CSL Rept. DC-121, UILU-ENG-90-2226, UIUC.

[5] J. R. Perkins, C.Humes Jr., and P. R. Kumar. Distributed control of flexible manufacturing systems: stability and performance. *IEEE Transactions on Robotics and Autonomation*, 37(1):132–141, April 1994.

[6] J. R. Perkins and P. R. Kumar. Stable distributed real-time scheduling of flexible manufacturing/assembly/disassembly systems. AC-34(2):139–148, February 1989.

[7] T.I.Seidman. private communication, 1994.

The PPS®-2000 Personal Communications System: Description and Performance Prediction Techniques and Results

Roger Peterson, Kevin Cutts, and John Haug
Communication Systems Research Laboratory
Motorola Inc
1301 E. Algonquin Road
Schaumburg, IL 60196

Introduction & Abstract

PPS®-2000 is a Personal Communications System (PCS) being developed by Motorola for use in the newly allocated spectrum near 1900 MHz. The system air interface is designed to support low mobility (non-vehicular) portable and fixed wireless access applications. This paper provides a high-level description of the system and one of the system performance estimation tools used in its evaluation. A critical figure of merit for the system is location reliability. The tool used to estimate location reliability is a static simulator which, after constructing a snapshot model of the system, calculates the carrier-to-interference power ratio for all system communication links. Location reliability is a single sample point of a statistical description of these C/(I+N) results. Although a brief description of the static simulator is provided, the primary goal of this paper is to discuss several concepts related to static simulations which have not previously been presented in the literature. Finally, the isolated cell location reliability calculation first presented in Jakes [1] is extended to microcellular path loss models.

PPS®-2000 System Description

The PPS®-2000 Personal Communication System is based on the Personal Access Communication System (PACS) air interface being defined in the Joint Technical Committee for Air Interfaces (JTC) which is sponsored by the Telecommunications Industry Association (TIA) and the Alliance for Telecommunication Industry Society (ATIS). PPS®-2000 is a TDMA radio system which is intended to provide service to pedestrian portable telephone users in a spectrally efficient manner with low cost equipment. Microcellular operation was the target environment in the design of this air interface. It is expected to be deployed with low transmitter power and low antenna heights.

A detailed description of the PACS air interface used by PPS®-2000 may be found in [2] and WACS is described in [3] which covers many of the system aspects beyond the air interface specification itself. Key system parameters are presented in Table I. A brief overview of the key features which are relevant to this simulation study are discussed below.

PPS®-2000 features a 300 kHz RF channel spacing and a channel bit rate of 384 kbps. Each RF channel has 8 time slots and the frames repeat at a 400 Hz rate corresponding to a 2.5 millisecond frame duration. Each time slot consists of 120 bits of which eighty are used to carry user information. This user channel supports 32 kbps ADPCM voice or data services up to 28.8 kbps. Ten of the 120 bits are used to provide a slow channel for associated signaling and control. The eighty bits of the user (or fast) channel and the 10 bits of the slow channel are protected by a fifteen bit cyclic redundancy check (CRC) which is used for error detection. For the purposes of our link level simulations which were used to define target C/I ratios for the system simulations a word error occurred only when the CRC detected an error.

Because of the low fading rates experienced in this system coupled with the short time slot length, forward error correction and interleaving are not efficient means of error mitigation and therefore are not used. This permits a minimization of speech path delay. For example PPS®-2000 uses a 2.5 millisecond frame as compared to the framing and interleaving delays in excess of 20 msec for IS-54 and GSM.

Table I. PPS®-2000 System Parameters

System Parameter	Value
Frequency Band	1850 - 1990 MHz
Access Method	TDM Down Link / TDMA Up Link
Duplex Method	Frequency Division (FDD)
Modulation	$\pi/4$-DQPSK with SRRC Pulse Shaping
Bit Rate	384 kbps
Channel Spacing	300 kHz
Frame Length	2.5 ms (8 time slots)
Time Slot Duration	312.5 µs (120 bits)
Power Control	Uplink Only - 'Delta Modulated'
RP Frequency Assignment	Quasi-Static Autonomous

The system features a Time Division Multiplexed (TDM) down link and a Time Division Multiple Access (TDMA) up link. Each of the eight RF channel time slots can be used as a traffic channel; one time slot per coverage area is reserved for system control. Powered subscriber units that are in the idle mode of operation synchronize and listen to the control channel. This channel provides its coverage area with operations information, alerting information and priority blocked call handling. Control channel protocols permit a discontinuous, or sleep mode, of operation in which idle mode subscriber units need to listen to the control channel only 1% of the time which significantly extends battery life. The use of time division multiplexing (TDM) on the down link implies that the radio ports (or base stations) are always transmitting. The portable units take advantage of this to implement very efficient diversity, system acquisition and handoff algorithms. Furthermore the radio ports themselves can take advantage of this to implement a self-organizing frequency assignment algorithm which is described below and which was one of the central issues to be studied through these simulations.

There are several aspects of the PPS®-2000 personal communication system that proved to be of particular importance and had a profound influence on the system simulations we are describing in this paper: up link power control, handoff and the automatic radio port channel assignment procedure called QSAFA [4]. These are described in the following paragraphs.

Power Control: In a microcellular system designed for very high reliability, the transmit power of the portable units must be set to provide adequate power margin at the cell boundary for shadowing and fading. Since these high power levels are not required over most of the coverage area, many of the subscriber units transmit much more power than is necessary to provide adequate service. Excessive transmit power will generate unnecessarily high interference levels on the up link of cochannel and adjacent channel portables. This interference can be controlled by incorporating transmit power control in the portable units [5]. In this system a single bit in the down link burst directs the portable unit to raise or lower its transmit power by one step each frame. Thus the power control channel (PCC) can be compared to a delta modulated signal and the output power is the running accumulation of the power control commands. The algorithm used in the radio port control unit to set the PCC bit attempts to set the power of the subscriber unit at a level somewhat above that

required for a telephone quality connection. One possible algorithm for setting the power control bit has been described by Bellcore [3,5].

Handoff: To maintain a high quality radio link, portable units continually monitor the radio environment, even during a call, to make sure that they are operating on the best possible channel. If a sufficiently better channel is found, the portable unit can execute an automatic link transfer (ALT) or a time slot transfer (TST) to begin operation on the better RF carrier or on a better time slot on the current RF carrier.

QSAFA: Microcells are used in PPS®-2000 to provide high network capacity with low power and low complexity radio ports. As these low power radio ports are deployed close together with low antenna heights, the frequency planning process quickly grows much more complex than that for a cellular system since it now depends much more heavily on the detailed structure of the physical environment surrounding both the radio port and the portable unit. System simulations have shown that an automatic frequency planning procedure performed by the radio ports can, on the average, provide as good a frequency plan as most manual planning methods [4]. In this approach each radio port periodically (on the order of once per day) turns off its transmitter and for a short time monitors all radio port transmit frequencies. It then chooses the frequency with the lowest detected interference power from all other ports and begins to transmit on the selected channel. This routine may be repeated several times and simulations have shown that after several iterations a stable frequency plan is reached that provides an adequate reuse pattern matched to the environment.

System Performance Metrics

Figures of merit which are critical to PCS include 1) location reliability, 2) blocking probability, and 3) dropped call rate. "Location reliability" is the probability that a subscriber at an arbitrary location in the system is provided with adequate communications quality. For this study "adequate communications quality" is achieved if the carrier-to-interference power ratio, C/(I+N), is larger than a threshold determined by link level analysis and simulations. "Blocking probability" is the probability that a subscriber will be refused service because all voice channels capable of providing adequate communications quality are being used by other subscribers. "Dropped call rate" is the probability that a call will be prematurely terminated by the system. A primary goal of the service provider is to assure that the system provides adequate service in terms of these figures of merit with the minimum investment in infrastructure equipment.

The concept of location reliability may be applied to a fully loaded cellular PCS or to an isolated cell having a single port and no co-channel interference. The calculation of isolated cell location reliability provides the system designer with a reasonable test of the top-level system design parameters such as maximum and minimum transmit power, receiver sensitivity, power control range, and so on. Isolated cell location reliability may be calculated analytically or using system simulators as discussed in the following paragraphs. The analytical calculation of isolated cell location reliability provides a means of checking the accuracy of the simulation tools.

Isolated cell location reliability is calculated in Jakes [1, p 126] for a system where path loss $L_p(d)$ may be represented by $L_{p1}(d) = K + A\log_{10}(d)$. In this case location reliability is [1]

$$LR(A,m) = \tfrac{1}{2}\left\{1 - erf(a) + \exp\left(\frac{1-2ab}{b^2}\right)\left[1 - erf\frac{1-ab}{b}\right]\right\} \qquad (1)$$

where $a = -m/\sqrt{2}\sigma$ and $b = A\log_{10}e/\sqrt{2}\sigma$. This result is not a function of the cell radius but is a function of the log-normal path loss margin at the cell edge and the path loss constant A. This relationship is based upon the assumption that the log-normal path loss from the port to any point in the system may be modeled by an independent draw of a zero mean Gaussian random variable with standard deviation σ. The log-normal path loss margin, m, at the cell edge is the difference between the receiver sensitivity and the received power at the cell edge in the absence of log-normal shadow fading.

The path loss model $Lp1(d)$ has been widely used for system studies of macrocellular mobile radio systems where the base antennas are placed higher than the average height of any buildings near the base stations. Microcellular port antennas are expected to be located below the average height of nearby buildings and a more complex path loss model must be used. Specifically,

$$
\begin{aligned}
L_{p2}(d) &= K + A\log_{10}(d) && d \le \frac{d_{bp}}{2} \\
&= K + A\log_{10}\left(\frac{d_{bp}}{2}\right) + B\log_{10}\left(\frac{2d}{d_{bp}}\right) && \frac{d_{bp}}{2} < d \le 2d_{bp} \\
&= K + A\log_{10}\left(\frac{d_{bp}}{2}\right) + BLog_{10}(8) + C\log_{10}\left(\frac{d}{4d_{bp}}\right) && 2d_{bp} \le d
\end{aligned}
\qquad (2)
$$

where d_{bp} is a function of port and subscriber unit antenna heights and the carrier wavelength, has been proposed [6] as a model appropriate for PCS microcellular systems. With path loss modeled by (2), Jakes' equation (1) must be extended. This calculation is straightforward but tedious. For cell radius $R \ge 2d_{bp}$, the result is

$$
LR(A,B,C,m,R) = I\left(\frac{d_{bp}}{2},a_1,b_1\right) + I\left(4d_{bp},a_2,b_2\right) - I\left(\frac{d_{bp}}{2},a_1,b_2\right) + I\left(R,a_3,b_3\right) - I\left(4d_{bp},a_2,b_3\right) \qquad (3)
$$

where

$$
I(r,a,b) = \frac{r^2}{2R^2}\left\{1 - erf(a) + \exp\left(\frac{1-2ab}{b^2}\right)\left[1 - erf\left(\frac{1-ab}{b}\right)\right]\right\}
$$

$$
a_1 = \left\{-m - B\log_{10}8 - C\log_{10}(R/4d_{bp})\right\}/\sqrt{2}\sigma
$$

$$
a_2 = \left\{-m - C\log_{10}(R/4d_{bp})\right\}/\sqrt{2}\sigma \qquad a_3 = -m/\sqrt{2}\sigma
$$

$$
b_1 = A(\log_{10}e)/\sqrt{2}\sigma \qquad b_2 = B(\log_{10}e)/\sqrt{2}\sigma \qquad b_3 = C(\log_{10}e)/\sqrt{2}\sigma
$$

This result is a function of the cell radius and log-normal path loss margin as well as the details of the path loss model. Similar results can be derived for $R \le 2d_{bp}$.

System Performance Prediction Using Static Simulators

Static simulators are system level performance prediction tools used by PCS service providers and manufacturers to estimate the quality of service to subscriber as a function of many system design decisions. Location reliability and blocking probability can be estimated using static simulators. Estimation of dropped call rate requires the use of dynamic simulators (not discussed here). Static simulators have been used since the first days of cellular telephony. As computer platforms have become increasingly powerful, static

simulators have evolved to include more features and capability. The following paragraphs describe the concepts used in static simulators developed by Motorola's Corporate Communication Systems Research Laboratory to predict performance of PPS®-2000. The specific techniques described here are applicable only to systems using narrowband modulation and must be modified for application to wideband spread-spectrum PCS.

At any instant in time a large number of subscribers are located throughout the system and each is served by and controlled by a port which was selected according to the call initialization and maintenance processes. Each subscriber experiences a communications quality which varies as a function of time due to the subscriber's mobility and due to the time variations in interference as other users calls start and stop and as other subscriber's locations change. The ALT and TST processes attempt to maintain the best possible call quality for all subscribers by serving each subscriber with the port and time slot requiring the minimum up link and down link transmit powers thus minimizing the interference generated by each subscriber. For the up link, transmit power control is also used to minimize transmit power and thus interference. The function of a static system simulator is to construct a time snapshot of the system so that call quality for all subscribers may be measured and statistics of call quality may be estimated. The simulated construction and loading of the system must take ALT, TST, power control, call blocking, subscriber unit placement, the system frequency plan, and all detail of the propagation environment into account.

The static simulator constructs the PCS model and evaluates the statistics of C/(I+N) by performing the following steps:

1) **Construct the log-normal path loss grid**. The path loss grid covers the entire coverage area. Grid point separation is equal to the distance between geographic points assumed to experience independent log-normal fading. Independent Gaussian random variables are generated and assigned to all grid points and all grid point pairs (see later discussion for detail) to represent log-normal path loss.

2) **Place ports.** Overlay the coverage area with the usual pattern of hexagons. Ports are placed at the center of the hexagons which approximately represent the coverage area for a single port. Port location error may be modeled by adding an appropriate error vector to the port location.

3) **Assign a carrier frequency to each port.** Carrier frequencies are assigned either manually or by modeling the QSAFA process. A single channel is assigned to each port. Manual planning separates co-channel ports by the maximum geographic distance. QSAFA frequency planning is modeled by letting one port at a time measure received power on all carrier frequencies and choosing the frequency with the minimum received power for transmission.

4) **Place subscribers.** Subscribers units are randomly placed within the boundary of the hexagon defining each cell. The probability density function for random placement is assumed to be uniform over the hexagon.

5) **Identify each subscriber as active or inactive.** Each subscriber unit is identified as active or inactive by random selection in accordance the system loading being modeled. Offered load is specified in Erlangs. If M subscribers are placed in each cell and the offered load is C Erlangs, a subscriber is active with probability C/M.

6) **Assign each active subscriber to a port.** Each subscriber measures the down link received power on each channel and constructs an ordered list of the three strongest channels. Perform the following steps for each active subscriber:

6.1) **Ports select best time slot.** Each port measures the up link received power on each time slot of the port's assigned channel and records the number of the time slot with the minimum received power.

6.2) **Assign subscriber to port.** If a time slot is available, assign subscriber to the geographically closest port which is using the strongest channel measured by the subscriber. Otherwise, if a time slot is available, assign subscriber to closest port which is using the second strongest channel measured. Otherwise, if a time slot is available, assign subscriber to closest port which is using the third strongest channel measured. If no channel is available on any of the three strongest channels, the subscriber is blocked.

7) **Execute up link power control.** Perform the following steps for each active subscriber:

7.1) Port measures up link C/(I+N)

7.2) **Adjust subscriber transmit power.** If the measured up link C/(I+N) is greater than threshold CIR_{high}, decrease subscriber unit transmit power by D. If the measured up link C/(I+N) is less than threshold CIR_{low}, increase subscriber unit transmit power by D. Assure that transmit power is between its high and low limits.

8) **Collect statistics of C/(I+N)** Measure and record the up link and down link C/(I+N) for every active subscriber in the system.

The entire procedure just described is executed many times in order to gather statistically significant results.

System performance is strongly a function of the propagation environment. System simulators may use either statistical models of the propagation environment or may use actual measured propagation data. When statistical models are used, the simulator's performance estimates do not apply to any specific PCS. When applying simulator results to specific systems, the system designer must use other tools to assure that propagation anomalies of the particular system of interest are accounted for properly. Statistical propagation models are used in this study. The propagation model used is a slight modification of the usual three component model including path loss denoted $L_P(d)$, log-normal or shadow fading denoted L_S, and multipath fading. Total path loss in decibels is the sum of these components. A link level simulation (not discussed here) accounts for multipath fading as well as all multipath mitigating techniques (e.g., transmit and receive diversity). The link simulation provides the minimum C/(I+N) denoted γ_{min} for acceptable link performance to the system simulation. This threshold C/(I+N) is a function of the microscopic multipath model being considered.

Many different path loss models have been proposed for evaluation of microcellular personal communication systems. More than a dozen path loss models have been implemented in the Motorola static simulator. Numerical results in this paper have been generated assuming the two-breakpoint three-slope path loss model of equation (2).

A three component log-normal or shadow fading model has been assumed. Total log-normal path loss between points A and B in decibels is $L_s(A,B) = L_{s1}(A) + L_{s2}(A,B) + L_{s1}(B)$. The first component $L_{S1}(A)$ accounts for propagation path obstructions near point A, the second component $L_{S2}(A,B)$ accounts for obstructions between the transmitter and receiver, and the third component $L_{S1}(B)$ accounts for obstructions near point B. All three components as well as their sum are zero mean Gaussian random variables. The variance σ_s^2 of L_S is

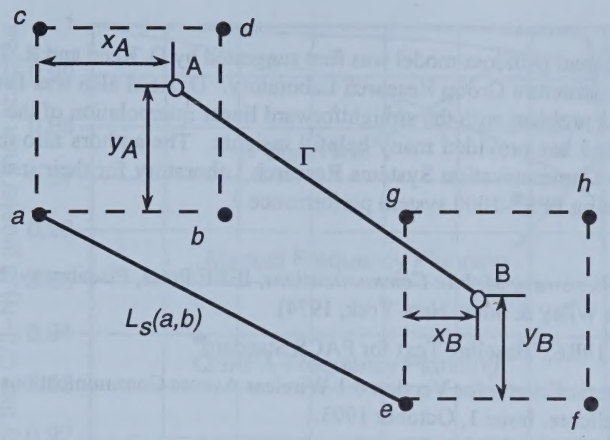
$\sigma_s^2 = 2\sigma_{s1}^2 + \sigma_{s2}^2$ where σ_{sj}^2 represents the variance of the individual components. The commonly used single-component log-normal shadow fading model is special case of this three-component model.

A very large number of calculations of log-normal path loss must be performed by the simulator. A single *independent* calculation of total log-normal path loss would involve the summation of three independent zero mean Gaussian random variables with the proper variance. Independent calculations of log-normal path loss do not, however, account for correlation between propagation paths which encounter the same physical obstructions. Multiple propagation paths which encounter the same or very similar obstructions will occur in the simulation when multiple subscribers are placed close to one another by the simulator's subscriber unit random placement process. In addition, when QSAFA is modeled, it is important that the log-normal path loss between two points calculated during the frequency planning procedure and the log-normal path loss calculated for the same two point as part of the calculation of C/(I+N) later in the simulation be identical. Correlation and repeatability in the calculation of log-normal path loss is achieved by defining a log-normal path loss grid covering the entire system. The spacing of grid points is equal to the distance between physical points in the system which are expected to experience independent log-normal fading. Figure 1 illustrates eight points in the log-normal path loss grid and the repeatable calculation of log-normal path loss between two points A and B surrounded by these grid points. The calculation is a modification of the interpolation method suggested in [4]. For this calculation $L_{S1}(a)$ and $L_{S1}(b)$ represent the log-normal fading due to transmission obstructions *near* grid points a and b and, in the simulation, are independent draws of Gaussian random variables for each $a \neq b$. $L_{S2}(a,b)$ represents the log-normal fading due to transmission obstructions *between* grid points a and b and, in the simulation, is an independent draw of a Gaussian random variable for each distinct pair $(a,b) \neq (a',b')$. The enhancement to [4] is the inclusion of the denominator term in the equation defining $\Gamma(A,B)$ in Figure 1. Without the denominator the calculation would be

$$\Gamma'(A,B) = \{\alpha[1-x_A] + \beta x_A\}\{1-y_A\} + \{\gamma[1-x_A] + \xi x_A\} y_A \qquad (2)$$

where all terms have been defined in Figure 1. This calculation is a straightforward linear interpolation or weighted average of sixteen different values of $L_s(a,b)$. The weighting of this average is a function of the location of A and B with respect to the reference grid. Given that the variance of $L_s(a,b)$ is σ_s^2, the variance of $\Gamma'(A,B)$ is a function of the locations (A,B). The variance of $\Gamma'(A,B)$ is less than the variance of $L_s(a,b)$ for any (A,B) not coincident with a grid point pair (a,b). The denominator in the equation for $\Gamma(A,B)$ in Figure 1 is a correction to force the variance $\Gamma(A,B)$ to equal the variance of $L_s(a,b)$. The correction term may be found by calculating the variance of $\Gamma'(A,B)$ for a fixed pair (A,B) and observing that this result is the product of the variance of $L_s(a,b)$ and a function $f(x_A, y_A, x_B, y_B)$. The correction term is $[f(x_A, y_A, x_B, y_B)]^{-1/2}$.

Figures 2 through 4 illustrate typical results obtained from the static simulator. The abscissa in both figures is C/(I+N) in decibels and the ordinate is the probability that a randomly placed subscriber will experience the abscissa C/(I+N) or better. For the C/(I+N) equal to threshold, this probability is equal to location reliability. Results are shown for both the up link and the down link. For example, the down link threshold C/(I+N) for the current PPS®-2000 hardware is +17 dB. From Figure 2 the down link location reliability is 94.6% for the conditions of that simulation run and a cell radius of 1000 feet; transmit powers used to generate Figure 2 were selected for a target cell radius of 1000 feet. Figure 3 illustrates the



$$\Gamma = \frac{\{\alpha[1-x_A]+\beta x_A\}[1-y_A]+\{\gamma[1-x_A]+\xi x_A\}y_A}{\sqrt{(1-2x_A+2x_A^2)(1-2x_B+2x_B^2)(1-2y_A+2y_A^2)(1-2y_B+2y_B^2)}}$$

$$\alpha = \{L_s(a,e)[1-x_B]+L_s(a,f)x_B\}[1-y_B]+\{L_s(a,g)[1-x_B]+L_s(a,h)x_B\}y_B$$

$$\beta = \{L_s(b,e)[1-x_B]+L_s(b,f)x_B\}[1-y_B]+\{L_s(b,g)[1-x_B]+L_s(b,h)x_B\}y_B$$

$$\gamma = \{L_s(c,e)[1-x_B]+L_s(c,f)x_B\}[1-y_B]+\{L_s(c,g)[1-x_B]+L_s(c,h)x_B\}y_B$$

$$\xi = \{L_s(d,e)[1-x_B]+L_s(d,f)x_B\}[1-y_B]+\{L_s(d,g)[1-x_B]+L_s(d,h)x_B\}y_B$$

$$L_s(a,b) = L_{s1}(a)+L_{s2}(a,b)+L_{s1}(b)$$

Figure 1. Log-normal path loss interpolation

degradation in location reliability due to using QSAFA rather than manual frequency planning. Figure 4 illustrates the distortion of location reliability results due to the use of the incorrect direct linear interpolation to calculate log-normal path loss. Observe in Figure 4 that the incorrect direct linear interpolation results in highly optimistic results.

Conclusions

Static system simulation is an extremely valuable PCS design tool. Static simulations can predict location reliability as well as call blocking rate as a function of many system design decisions. Motorola is using static simulations to estimate the maximum and minimum cell size for PCS as well as to evaluate handover and power control algorithm options. Although static simulation is useful, it is limited in its ability to evaluate system control processes which interact rapidly with the dynamic PCS propagation environment. For this reason, Motorola is also developing dynamic PCS simulators. Finally, simulation results are critically dependent upon the assumptions made regarding the propagation environment.

Acknowledgments

The three component log-normal path loss model was first suggested by D. Reed and R. Tang of Motorola's Cellular Infrastructure Group Research Laboratory. D. Reed also was first to point out the possibility of a problem with the straightforward linear interpolation of the log-normal random variables and has provided many helpful insights. The authors also thank their colleagues within the Communication Systems Research Laboratory for their insights into the critical issues affecting PPS®-2000 system performance.

References

[1] W. C. Jakes, editor, *Microwave Mobile Communications,* IEEE Press, Piscataway, NJ: 1994 (originally: John Wiley & Sons: New York, 1974).

[2] JTC(AIR)/94.08.30-119R6, "Baseline Text for PACS Standard"

[3] TR-INS-001313, Generic Criteria for Version 0.1 Wireless Access Communications Systems (WACS), Bellcore, Issue 1, October 1993.

[4] Justin C-I. Chuang, "Autonomous Adaptive Frequency Assignment for TDMA Portable Radio Systems," *IEEE Transactions on Vehicular Technology*, August 1991.

[5] Justin C-I. Chuang and Nelson R. Sollenberger, "Uplink Power Control for TDMA Portable Radio Channels," *IEEE Transactions on Vehicular Technology*, February 1994.

[6] JTC(AIR)/94.08.01-065R4, "Draft Technical Report on RF Channel Characterization and System Deployment Modeling".

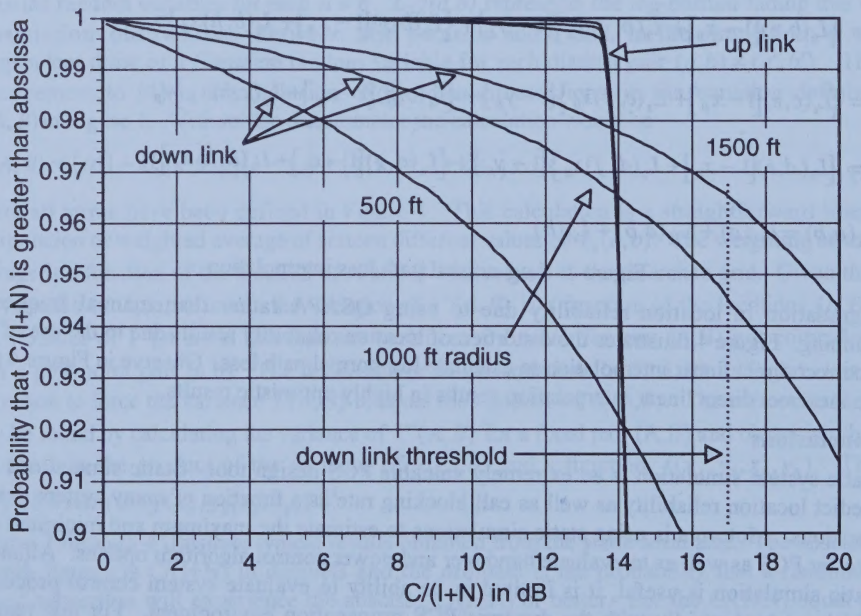


Figure 2. Static simulator output for cell radii of 500, 1000, and 1500 feet.

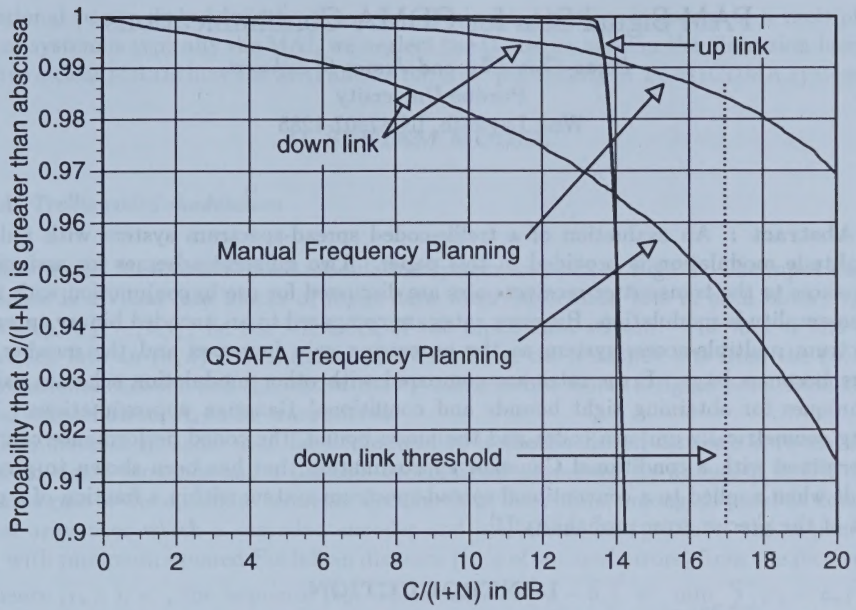


Figure 3. Static simulator output for a cell radius of 1000 feet with manual and QSAFA frequency planning.

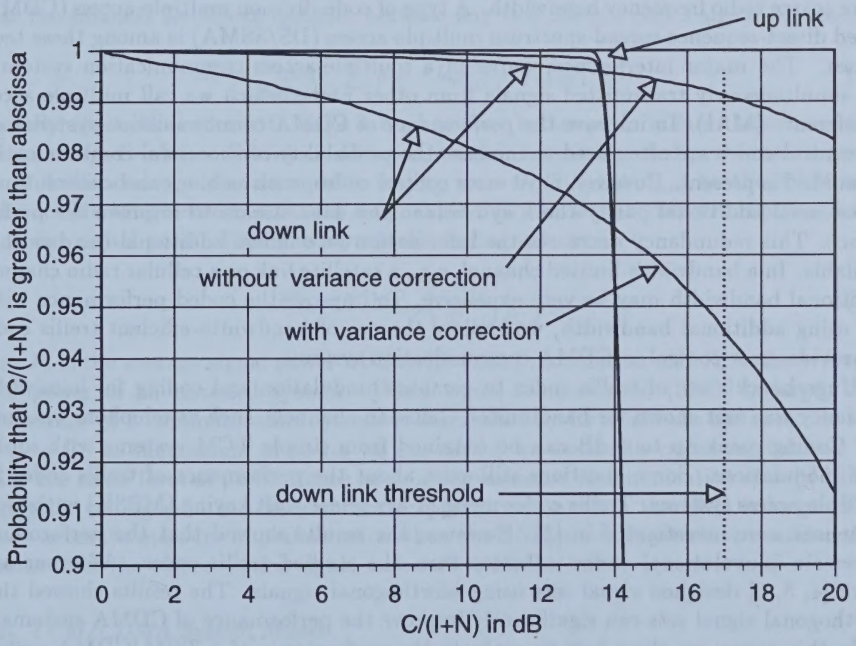


Figure 4. Static simulator output for a cell radius of 1000 feet using correct and incorrect log-normal interpolation models.

PAM Signal Sets for CDMA Communications

Hsiao-Chiu Chu and James S. Lehnert
Purdue University
West Layfette, IN 47907-1285

Abstract : An evaluation of a trellis-coded spread-spectrum system with pulse-amplitude modulation is provided in this paper. Two different schemes for assigning sequences to the transmitter-receiver pairs are discussed for use in conjunction with the pulse-amplitude modulation. Bit error rates are compared to an uncoded binary spread-spectrum multiple-access system as the processing gain increases and the number of users becomes large. Error rates are compared with other modulation schemes using techniques for obtaining tight bounds and conditional Gaussian approximations. By using geometrically uniform codes and the union bound, the coded performance can be determined with a conditional Gaussian approximation that has been shown to give a result when applied to a conventional spread-spectrum system within a fraction of a per cent of the precise error probability [1].

I. INTRODUCTION

During recent years, various packet radio techniques have been proposed to efficiently share scarce radio frequency bandwidth. A type of code-division multiple-access (CDMA) called direct-sequence spread-spectrum multiple-access (DS/SSMA) is among these techniques. The major interference source in a multiple-access communication system is the simultaneously transmitted signals from other users, which we call multiple-access interference (MAI). To improve the performance of CDMA communication systems, error control codes are often used to increase the probability of successful communication when MAI is present. However, most error control codes, such as block and convolutional codes, need additional parity check symbols in the data stream to improve the performance. This redundancy decreases the information rate unless additional bandwidth is available. In a bandwidth-limited channel, e.g., a satellite link or a cellular radio channel, additional bandwidth may be very expensive. To improve the coded performance without using additional bandwidth, we explore the use of bandwidth-efficient trellis codes to provide error control in CDMA communication systems.

Ungerboeck's use of trellis codes to combine modulation and coding for bandwidth efficiency was first shown for bandlimited Gaussian channels, such as telephone channels [2]. Coding gains up to 6 dB can be obtained from simple TCM systems with multilevel modulations. Some questions still exist about the performance of trellis codes for multiple-access systems. Trellis codes using M-ary phase shift keying (MPSK) with Gold sequences were investigated in [3]. However, the results showed that the performance of certain convolutional codes is better than the studied trellis codes. Woerner and Stark [4, 5, 6] designed signal sets using biorthogonal signals. The results showed that biorthogonal signal sets can significantly improve the performance of CDMA systems.

In this paper, we show how to evaluate the performance of a TCM/CDMA system using pulse amplitude modulation. In other work, we have evaluated a new type of signal set that combines PAM with several orthogonal direct sequences. Both systems are capable of improving the performance of a CDMA communication system without

additional power or bandwidth. Since the dominant interference source of a multiple-access system is typically the MAI, we neglect the Gaussian noise in the discussion below to give a clear picture how the MAI affects the error probability of TCM/CDMA systems.

II. SYSTEM MODEL

A. Trellis-coded modulation

The model for a trellis-coded modulation system is shown in Fig. 1. The input data sequence is divided into blocks of input data with $r_b$user data bits in each block. Let $b_i = (b_{i,1}, b_{i,2} \ldots, b_{i,r_b})$ be the ith block of the input data stream. The first k bits in b_i are fed into an (n, k) convolutional encoder. Then the n output bits from the (n, k) convolutional encoder and the rest of the $r_b - k$ data bits select a signal point from the signal constellation, A, to be transmitted.

The distance measure used for demodulation is the Euclidean distance between the received signal sequence and a possible coded signal sequence. As discussed in [7], the decision rule of the optimum sequence decoder is to determine, among all possible coded signal sequences which a cascaded encoder and modulator can produce, the sequence $(\tilde{b}_n)$ with minimum squared Euclidean distance (sum of squared errors) from the received sequence (r_n); i. e. , the sequence $(\tilde{b}_n)$ which satisfies $\left|r_n - \tilde{b}_n\right|^2 = \min_{\{a_n\} \in C} \sum |r_n - a_n|^2$. The objective of designing a trellis code is to separate distinct coded signal sequences with large Euclidean distance. We define the free Euclidean distance of a trellis code, d_f, as the smallest Euclidean distance between any two distinct coded signal sequences; i. e. ,

$$d_f^2 = \min_{\{a_n\} \neq \{b_n\}} \sum |a_n - b_n|^2; \{a_n\}, \{b_n\} \in C \tag{1}$$

Generally, the trellis codes with large $d_f$tend to have better performance than the codes with small d_f. From free Euclidean distance, another frequently used performance measure of trellis codes is the asymptotic coding gain G_C[7]. Let d_{min} be the minimum Euclidean distance between points of a signal set that can be encoded with up to $r_b$data bits in one signal point. Then the asymptotic coding gain is:

$$G_C = 10 \log_{10} \left(\frac{d_f^2 / P_c}{d_{min}^2 / P_u} \right), \tag{2}$$

where P_c is the average signal power with trellis coded modulation and P_u is the average signal power of an uncoded system. When signal-to-noise ratio (SNR) is large, the performance of a communication system is dominated by its minimum distance. So we can view G_C as the amount of power we can save if we use trellis coded modulation to obtain the same level of performance as in a conventional communication system. The asymptotic coding gain is a very accurate approximation when the SNR is large. For a more accurate measure of performance when the SNR is small, we must consider the distance from the correct coded signal sequence to all the other sequences in C.

B. TCM/CDMA System Model

The model of the TCM/CDMA communication system under consideration is shown in Fig. 2. It is similar to the models defined in [8] and [9] except for additional modules of

TCM encoders. The trellis encoder uses a block of r_b data bits to choose the transmitted signal from a signal constellation. The received signal in this asynchronous CDMA system is the sum of K spread-spectrum signals $s_k(t-\tau_k), 1 \leq k \leq K$, plus an additive white Gaussian noise (AWGN) process $n(t)$, which has two-sided spectral density $N_0/2$. The delayed spread-spectrum signal for user k is given by

$$s_k(t-\tau_k) = \sqrt{2P}\tilde{b}_k(t-\tau_k)a_k(t-\tau_k)\cos(\omega_c t + \phi_k), \tag{3}$$

where $\tilde{b}_k(t)$ is a process resulting from the trellis code and input data signal $b_k(t)$, $a_k(t)$ is the spectral- spreading signal, τ_k is the time delay parameter that accounts for propagation delay and the lack of synchronism between the transmitters, ϕ_k is the phase angle of the kth carrier, and P is the power of the transmitted signal.

If a rectangular pulse is defined by $p_T(t) = 1$ for $0 \leq t \leq T$ and $p_T(t) = 0$, otherwise, then the process $\tilde{b}_k(t)$ can be expressed as $\tilde{b}_k(t) = \sum_{l=-\infty}^{\infty} \tilde{b}_l^{(k)} p_T(t - lT)$, where the sequence $(\tilde{b}_l^{(k)})$ is the output sequence from the trellis encoder of the kth transmitter ($\tilde{b}_l^{(k)} \in \{\pm(M-1), \pm(M-3), \ldots, \pm 3, \pm 1\}$ for each l). If the chip waveform $\psi(t)$ is defined to be time-limited to the interval $[0, T_c)$ and normalized to have energy T_c, the spectral-spreading signal of the kth transmitter can be expressed as $a_k(t) = \sum_{j=-\infty}^{\infty} a_j^{(k)} \psi_T(t - lT)$, where $(a_j^{(k)})$ is a periodic binary sequence of elements from the set $\{-1, +1\}$. We assume that each element in the sequence $(\tilde{b}_l^{(k)})$ is encoded with N chips; i. e. , $T = NT_c$, that the signature sequence $(a_j^{(k)})$ has period N, and that the chip waveform is rectangular.

As for the spectral-spreading sequences, we assume that these direct sequences are randomly generated and the choice of a sequence for each transmitter-receiver pair is made independently of the choices made for all other pairs. There are two different ways to assign direct sequences in our study. One is using a unique N-chip direct sequence for each transmitter/receiver pair. That is, for each data interval, the output amplitude modulates the same N-chip signature sequence. Another method is by assigning each transmitter/receiver pair several randomly generated N-chip direct sequences and using these direct sequences sequentially. For example, if each user is assigned L direct sequences, then in the first interval, the transmitter is using direct sequence one as the spectral-spreading sequence; in the second interval, the transmitter is using direct sequence two; etc. After all direct sequences have been used once, the transmitter starts over with the direct sequence one in the next data interval. The generation of the direct sequences is totally random and independent of the operation of the trellis encoder or the input data. The reason for using several N-chip direct sequences in cascade is to decrease the correlation between the MAI of adjacent data intervals. We also assume that the trellis codes used in all pairs share the common M-ary signal set, $\{\pm(M-1), \pm(M-3), \ldots, \pm 3, \pm 1\}$. The average probability of error that we obtain is an average with respect to all the possible combinations of signature sequences that might be selected.

We use correlation receivers to demodulate the received signal. The input signal is correlated with a unit-magnitude signature sequence. The decision circuit provides a sequence of decision statistics for the maximum-likelihood Viterbi decoder. The Viterbi decoder uses all decision statistics to form an estimate of the transmitted coded signal sequence. The decision statistic for the 1st user at interval l correlated by the signature sequence $(a_j^{(1)})$ is

$$\mathbf{Z}_l^{(1)} = T\sqrt{P/2}\left[\tilde{b}_l^{(1)} + \sum_{k=2}^{K} I_{k,1}(\tilde{b}_{k,l}, \tau_k, \phi_k)\right] \tag{4}$$

where $I_{k,1}(\tilde{b}_{k,l}, \tau, \phi) = \frac{1}{T} B_{k,1}(\tilde{b}_{k,l}, \tau) \cos\phi$ and $B_{k,1}(\tilde{b}_{k,l}, \tau) = \tilde{b}_{l-1}{}^{(k)} R_{k,1}(\tau) + \tilde{b}_l{}^{(k)} \hat{R}_{k,1}(\tau)$. The definitions of $R_{k,1}(\tau)$ and $\hat{R}_{k,1}(\tau)$ are given in [8].

For an uncoded system, the decoder just selects the amplitude which is closest to the decision statistic as the transmitted amplitude. For a TCM system, all decision statistics are input to a soft-decision Viterbi decoder to determine the most likely coded signal sequence. The Viterbi decoder selects the sequence closest to the received signal in squared Euclidean distance as the output. In other words, a coded signal sequence p is selected when

$$\sum_{l=-\infty}^{i-1} \left(\tilde{b}_{p,l} - \frac{\mathbf{Z}_l}{T\sqrt{P/2}} \right)^2 < \sum_{l=-\infty}^{i-1} \left(\tilde{b}_{q,l} - \frac{\mathbf{Z}_l}{T\sqrt{P/2}} \right)^2 , \forall q \in C \text{ and } q \neq p \qquad (5)$$

where $\tilde{b}_{p,l}, \tilde{b}_{q,l}$ are corresponding elements of sequences p and q during the lth signaling interval.

III. PERFORMANCE ANALYSES

The TCM/CDMA system in this study has a PAM signal set. The transmitted signal from the kth user is the signature sequence with PAM amplitudes. We will focus on the error performance of the first user.

We begin our analysis by calculating the squared Euclidean distance from the received sequence to the correct sequence p and an error event q , where the Euclidean distance between p and q is d_f. The expression for the decision statistic for different time intervals is shown in (4). To simplify our discussion, we assume $P = 1$. Then we can write down the Euclidean distance between the received sequence and the correct sequence p and the Euclidean distance between the received sequence and the error event q as

$$S_p^{(1)} = \sum_{l=1}^{L_q} [\tilde{b}_{p,l}^{(1)} - \frac{\sqrt{2}}{T} Z_l^{(1)}]^2 = \sum_{l=1}^{L_q} [\tilde{b}_{p,l}^{(1)} - (\tilde{b}_{p,l}^{(1)} + \sum_{k=2}^{K} I_{k,l}^{(1)})]^2 = \sum_{l=1}^{L_q} [\sum_{k=2}^{K} I_{k,l}^{(1)}]^2 ,$$

$$S_q^{(1)} = \sum_{l=1}^{L_q} \left[\tilde{b}_{q,l}^{(1)} - \frac{\sqrt{2}}{T} Z_l^{(1)} \right]^2 = \sum_{l=1}^{L_q} \left[\tilde{b}_{q,l}^{(1)} - (\tilde{b}_{p,l}^{(1)} + \sum_{k=2}^{K} I_{k,l}^{(1)}) \right]^2 \qquad (6)$$

In (6), $\tilde{b}_{p,l}^{(1)}$ is the output amplitude in the lth interval of sequence p and $I_{k,l}^{(1)} = I_{k,1}(\tilde{b}_{k,l}, \tau_k, \phi_k)$. Errors will occur when the received sequence is closer to the error event than the correct sequence, i. e. , $S_p^{(1)} > S_q^{(1)}$. To simplify notation, we remove the superscript. Now, the difference between S_q and S_p can be written as

$$S_q - S_p = \sum_{l=1}^{L_q} \left[(\tilde{b}_{q,l} - \tilde{b}_{p,l})^2 - 2(\tilde{b}_{q,l} - \tilde{b}_{p,l}) \sum_{k=2}^{K} I_{k,l} \right]. \qquad (7)$$

It is important to notice that the free Euclidean distance in this type of TCM/CDMA system is determined by the signal amplitudes only. We can express the free Euclidean distance as

$$d_f^2 = \sum_{l=1}^{L_q} [\tilde{b}_{q,l} - \tilde{b}_{p,l}]^2. \qquad (8)$$

If we define W_k as the weighted sum of the MAI from the kth user, expressed as

$$W_k = \sum_{l=1}^{L_q} \left[(\tilde{b}_{q,l} - \tilde{b}_{p,l}) I_{k,l} \right], \text{ then } S_q - S_p = d_f^2 - 2 \sum_{k=2}^{K} W_k. \tag{9}$$

To find the density function of W_k, we use a method similar to [9]. In [9], W_k is decomposed into a sum of conditionally independent random variables and most of the random variables are grouped into two sets. Our result is a little more complicated because of the non-binary amplitudes and the trellis code, so we group most of the random variables into four sets. A derivation and calculation of the density function is shown in [10], from which we write down the final representation of W_k as

$$W_k = \frac{1}{T} \sum_{l=1}^{L_q} \left\{ \left[(\tilde{b}_{q,l} - \tilde{b}_{p,l}) B_{k,1}(\tilde{b}_{l-1}^{(k)}, \tilde{b}_l^{(k)}, \tau_k) \right] \cos \phi_k \right\}, \tag{10}$$

where $B_{k,1}(\tilde{b}_{l-1}^{(k)}, \tilde{b}_l^{(k)}, \tau_k)$ is the sum of $N+1$ random variables weighted by amplitudes $\tilde{b}_{l-1}^{(k)}$ and $\tilde{b}_l^{(k)}$ The definition of $B_{k,1}(\tilde{b}_{l-1}^{(k)}, \tilde{b}_l^{(k)}, \tau_k)$ can be found in [10]. The calculation of the density function for W_k is very similar to the calculation of the MAI in an uncoded biphase CDMA communication system [9]. However, we must point out that the density function of total MAI we derive is conditioned on $C(1)$. From each conditional density function, we can upper and lower bounds of the bit error rate. To obtain the bit error rate when the sequences are generated randomly, we need to average over $C(1)$. For systems using only one signature sequence with M levels of amplitude for each user, the computational complexity needed to calculate the density function of the total MAI (if we compute the density function of total MAI using (10) by conditioning on $C(1)$) is $N(N+1)^4 M^{L_q}$, with N convolutions. For systems using cascaded direct sequences for each user, the computation complexity is $[N(N+1)^4 M]^{L_q}$.

IV. IMPROVED GAUSSIAN APPROXIMATION

As we discussed in the previous section, the computation complexity increases exponentially with L_q. If L_q becomes too large, the computations will take a very long time even for a small N. In this section, we will apply an improved Gaussian approximation [1] to obtain the performance of a TCM/CDMA system using cascaded direct sequences.

A. Variance of the MAI

To apply the improved (conditional) Gaussian approximation, we first need to find out the conditional variance Ψ of the MAI which is given by

$$\Psi \equiv Var[MAI \mid S_k, \phi_k, C(1), k = 2, 3, \ldots, K] = E[(\sum_{k=2}^{K} W_k)^2 \mid S_k, \phi_k, C(1), k = 2, 3, \ldots, K]$$
$$= \sum_{k=2}^{K} E[B_k^2 \mid S_k, C(1)] E[\cos \phi_k^2 \mid \phi_k] = \sum_{k=2}^{K} \frac{1}{2} [1 + \cos 2\phi_k] Var[B_k \mid S_k, C(1)], \tag{11}$$

, where $W_k = B_k \cos \phi_k$ for all k. Let $\gamma_k = \lfloor \frac{S_k}{T_c} \rfloor$. In sections IV-A and IV-B, the parameters α_l and γ_k will be used to represented the conditional variance of the MAI.

The parameter α_l is related to the direct sequence used in the lth interval, and it is always less than or equal to γ_k.

A.1. Single Signature Sequence Case

Since the same signature sequence is transmitted for all data intervals, we write α_l as α for all l. We need to further condition on the amplitudes used by user k and calculate the conditional variance of B_k as follows:

$$Var[B_k \mid S_k, C(1)] = E[B_k^2 \mid S_k, C(1)] = \sum_{\tilde{\mathbf{b}}^{(k)}} Prob\{\tilde{\mathbf{b}}^{(k)}\} E[B_k^2 \mid S_k, C(1), \tilde{\mathbf{b}}^{(k)}]. \quad (12)$$

To simplify the notation, we express the amplitudes of interfering signals from the kth user as the random vector $\tilde{\mathbf{b}}^{(k)} = (\tilde{b}_0^{(k)}, \tilde{b}_1^{(k)}, \ldots, \tilde{b}_{Lq}^{(k)})$. Now, the conditional variance of the MAI from the kth user can be calculated and found (for N even) as $E[B_k^2 \mid S_k, C(1), \tilde{\mathbf{b}}^{(k)}] =$

$$\sum_{\gamma_k=0}^{N-1} \sum_{\alpha=0}^{\alpha_{min}} P(\alpha, \gamma_k, C(1), N)\{[\tilde{\mathbf{B}}_1^2(\gamma_k - \alpha) + \tilde{\mathbf{B}}_2^2(\frac{N-1-C(1)}{2} - \gamma_k + \alpha)](1-2S)^2$$

$$+\tilde{\mathbf{B}}_1^2\alpha + \tilde{\mathbf{B}}_2^2(\frac{N-1+C(1)}{2} - \alpha))\} + \tilde{\mathbf{B}}_1^2 S^2 + \tilde{\mathbf{B}}_2^2(1-S)^2 + (-1)^{\frac{C(1)+1}{2}} \tilde{\mathbf{B}}_1 \tilde{\mathbf{B}}_2 S(1-S), \quad (13)$$

where $\tilde{\mathbf{B}}_1 = \frac{1}{T}\sum_{l=1}^{L_q}(\tilde{b}_{q,l} - \tilde{b}_{p,l})\tilde{b}_{l-1}^{(k)}$, $\tilde{\mathbf{B}}_2 = \frac{1}{T}\sum_{l=1}^{L_q}(\tilde{b}_{q,l} - \tilde{b}_{p,l})\tilde{b}_l^{(k)}$,

$\alpha_{min} = \min\{\frac{N-1+C(1)}{2}, \gamma_k\}$, *and*

$$P(\alpha, \gamma, C, N) = \begin{cases} 0, & \text{if } \frac{N-1}{2} + \alpha < \frac{C}{2} + \gamma \\ \frac{2^{1-N}}{N} \begin{pmatrix} \gamma \\ \alpha \end{pmatrix} \begin{pmatrix} N-1-\gamma \\ \frac{N-1+C(1)}{2} - \alpha \end{pmatrix}, & \text{otherwise.} \end{cases}$$

A.2. Cascaded direct sequence case

For the systems assigning each user several randomly-generated direct sequences, we choose L (the number of direct sequences assigned to each user) so that $L_q < L$ to minimize correlations of the MAI for consecutive signaling intervals. Then the direct sequence in each interval is independent of direct sequences used in adjacent data intervals. To simplify the notation, we express the amplitudes of the interfering signal from the kth user as the random vector $\tilde{\mathbf{b}}^{(k)} = (\tilde{b}_0^{(k)}, \tilde{b}_1^{(k)}, \ldots, \tilde{b}_{L_q}^{(k)})$ and the $C(1)$of the direct sequence used by the 1st user in each data interval as the random vector $C = (C_0, C_1, \ldots, C_{L_q})$. We need to further condition on the amplitudes used by user k and calculate the conditional variance of B_k as follows:

$$Var[B_k \mid S_k, C] = E[B_k^2 \mid S_k, C] = \sum_{\tilde{\mathbf{b}}^{(k)}} Pr\{\tilde{\mathbf{b}}^{(k)}\} E[B_k^2 \mid S_k, C, \tilde{\mathbf{b}}^{(k)}] \quad (14)$$

Now, the conditional variance of the MAI from the kth user can be calculated as in the previous subsection. The difference is that new sequences are used in each interval so the MAI of each interval must be calculated separately.

B. Variance computations

The final form for the variance of the conditionally independent MAI from each interfering user can be expressed as

$$\Psi_k = (1 + \cos 2\phi) \sum P(A, B, C)(AS^2 + BS + C) \tag{15}$$

, where $P(A, B, C)$is the probability of the combination (A,B,C). The conditional density function of each term in (15) is shown in [10]. After combining all possible density functions, we obtain the density function $f_{\Psi_k}(\psi_k)$ of the variance Ψ_k. We recall that Ψ_k are conditionally independent given C, where C is $C(1)$ for the case of a single signature sequence and $(C_0, C_1, \ldots, C_{L_q})$ for the case when sequences are cascaded. By performing $K-2$ convolutions of $f_{\Psi_k}(\psi_k)$ we obtain $f_{\Psi}(\psi)$, the density function of the total variance of the MAI conditioned on C. Now, from conditional density function $f_{\Psi}(\psi)$, we calculate the improved Gaussian approximation for the error probability conditioned on C ,$\hat{P}_{e,C}$, as

$$\hat{P}_{e,C} = \int_0^{\infty} Q(\frac{Nd_f^2}{\sqrt{\psi}}) f_{\Psi}(\psi) d\psi, \text{ where } Q(x) = \frac{1}{2\pi} \int_x^{\infty} e^{\frac{-u^2}{2}} du. \tag{16}$$

To find out the unconditional error probability, we just average over C to obtain

$$\hat{P}_e = \sum_{all\ C} P(C) \hat{P}_{e,C}. \tag{17}$$

V. NUMERICAL RESULTS

In this section, numerical results for simple TCM/CDMA systems with PAM are given to illustrate the performance improvement of TCM systems over traditional uncoded CDMA systems. All the figures in this section show the bit error rate versus the number of simultaneous users.

The structure of the encoder for this trellis code is shown in Fig. 3. This is a TCM system using a coded four amplitude system to replace an uncoded two amplitude system. From [2], we know the asymptotic coding gain of this trellis code is about 2.55 dB. For this trellis code, the number of input bits, r_b, is 1, so the processing gain is $N/r_b = N$, and each coded signal sequence has only one closest error event.

We show the numerical results for N=8, 16, and 32 in Figs. 4, 5, and 6. For each figure, the fine dash lines are the upper and lower bounds for the bit error rate of uncoded binary CDMA systems and the coarse dash lines are the upper and lower bounds for the bit error rate of TCM/CDMA systems using a single signature sequence. The solid lines are the improved Gaussian approximations to the bit error rate of both uncoded binary CDMA systems and TCM/CDMA systems using cascaded direct sequences. Since the MAI terms in different intervals are correlated with each other through the same phase, delay, and signature sequence used by decoder, we find the performance of a TCM/CDMA system using a single signature sequence does not achieve the asymptotic coding gain when the number of users is small compared to the processing gain. In fact, the improvement over uncoded CDMA systems deteriorates as processing gain increases. The reason for this phenomena is that the MAI for adjacent data intervals are highly correlated when the number of users is small. As the number of users increases, the correlation between

the MAI of adjacent data intervals decreases due to many different contributions to the interference. For the TCM/CDMA systems using cascaded direct sequences, the MAI between adjacent data intervals is only correlated by the phase, delay, and transmitted amplitudes of the desired user, so the degradation due to correlated MAI is much smaller compared to the systems using a single signature sequence.

VI. CONCLUSION

In this paper, we have examined the possibility of applying TCM techniques for CDMA communication systems. We designed TCM/CDMA systems with PAM and used several randomly generated direct sequences cascaded to optimize the error performance. As a result, we found that the asymptotic coding gains can be achieved for TCM/CDMA systems when the major interference is MAI.

References

[1] R. K. Morrow and J. S. Lehnert, "Packet throughput in slotted ALOHA DS/SSMA radio systems with random signature sequences", *IEEE Transactions on Communications*, vol. COM-40, pp. 1223–1230, July 1992.

[2] G. Ungerboeck, "Channel coding with multilevel/phase signals", *IEEE Transactions on Information Theory*, vol. IT-33, pp. 55–67, Jan. 1982.

[3] G. D. Boudreau, D. D. Falconer, and S. A. Mahmoud, "A comparison of trellis coded versus convolutionally coded spread-spectrum multiple-access systems", *IEEE Journal on Selected areas in Communications*, vol. 8, pp. 628–641, May 1990.

[4] W. E. Stark and B. D. Woerner, "The application of trellis coding techniques to direct-sequence spread-spectrum multiple-access communications", *in Proceedings of the Twenty-Sixth Annual Allerton Conference on Communications, Control, and Computing*, vol. 2, pp. 1070–1079, Sept. 1988.

[5] B. D. Woerner and W. E. Stark, "Trellis coded spread-spectrum for fading channels", *in Proceedings of 1990 IEEE Military Communications Conference*, pp. 389–393, Sept. 1990.

[6] B. D. Woerner, *Coded modulation for direct sequence spread spectrum communications*, PhD thesis, Univ. of Michigan, August 1991.

[7] G. Ungerboeck, "Trellis-coded modulation with redundant signal sets–part i & ii", *IEEE Communications Magazine*, vol. 25, pp. 5–21, Feb. 1987.

[8] M. B. Pursley, "Performance evaluation for phase-coded spread-spectrum multiple-access communication - Part I: System analysis", *IEEE Transanctions on Communications*, vol. COM-25, pp. 795–799, Aug. 1977.

[9] J. S. Lehnert and M. B. Pursley, "Error probabilities for binary direct-sequence spread-spectrum communications with random signature sequences", *IEEE Transanctions on Communications*, vol. COM-35, pp. 87–98, Jan. 1987.

[10] H. Chu, *Trellis-coded modualtion for CDMA*, PhD thesis, Purdue Univ., West Lafayette, IN, May 1994.

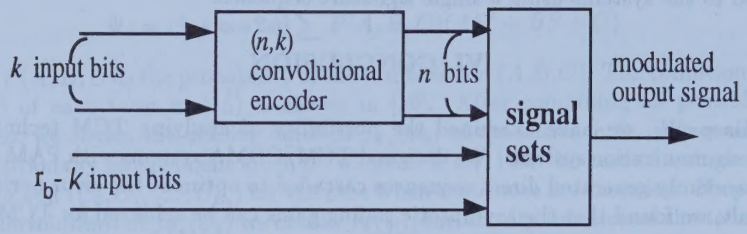


Figure 1: Fig. 1. System model for trellis coded modulation.

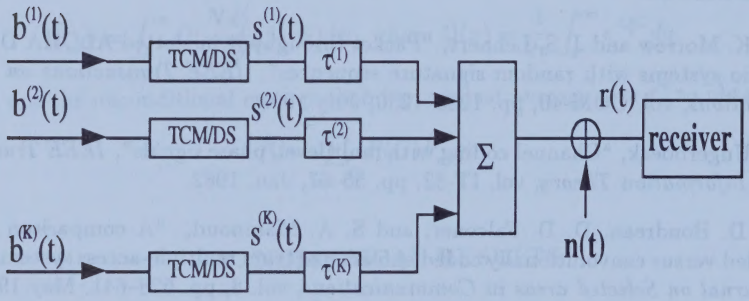


Figure 2: Fig. 2. TCM/CDMA communication system model.

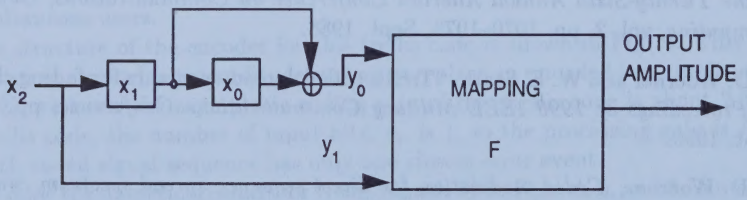


Figure 3: Fig. 3. Encoder for trellis code.

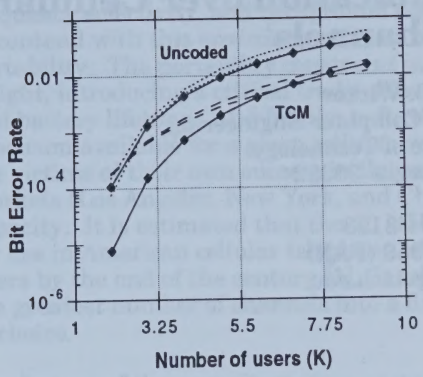


Figure 4: Bit error rate of a TCM/CDMA system using PAM compared to an uncoded binary CDMA system (processing gain =8).

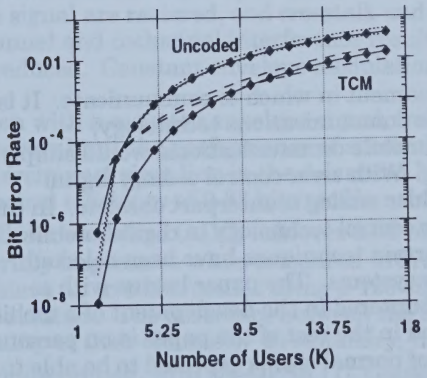


Figure 5: Bit error rate of a TCM/CDMA system using PAM compared to an uncoded binary CDMA system (processing gain =16).

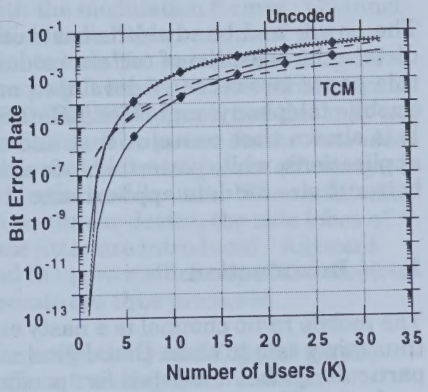


Figure 6: Bit error rate of a TCM/CDMA system using PAM compared to an uncoded binary CDMA system (processing gain =32).

Error Control Techniques for Voice and Data Communication over Cellular Mobile Channels[1]

Stephen B. Wicker
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332

(404) 894-3129
(404) 853-9959 (FAX)
wicker@ee.gatech.edu

Abstract

The power and bandwidth constraints that dominate mobile radio applications dictate the selection of certain modulation formats and error control systems. In this paper two recently developed mobile communication standards, ETSI/GSM (mobile telephony) and ETSI/ERMES (paging), are used to illustrate this process. It is shown that convolutional and trellis coded systems are preferred for voice applications, while powerful block codes, in particular Reed-Solomon codes, are the better choice for data applications.

1. Introduction

The mobile radio channel is a nasty environment in which to communicate. It is thus fairly safe to claim that digital mobile communications technology, particularly that developed for "personal" mobile communications, would simply not be feasible without error control coding. With error control coding, digital mobile radio is now rapidly rendering its elder analog counterpart obsolete. In this paper the author discusses the use of error control technology in digital mobile radio, and makes an effort to show why certain techniques have been selected over others in the design of various mobile systems. The paper begins with a discussion of the constraints typically encountered in the development of a mobile communication system. The focus here and in the rest of the paper is on personal, cellular communications - a human being of normal size is expected to be able to comfortably carry the portable unit in a brief case, purse, or even shirt pocket. The sections that follow consider the key differences between voice applications and data transfer applications. It is shown that these differences lead to the selection of different types of error control systems. A specific example if provided in each case: the European Telecommunications Standards Institute/Groupe Special Mobiles standard (ETSI/GSM) has been selected to represent cellular telephony, while the European Radio Messaging Service standard (ERMES) will represent cellular data transfer.

---

[1]The research in this report was supported in part by Mobile Telephone Technologies, Jackson, Mississippi.

2. General Constraints of the Mobile Application

The land mobile radio channel is corrupted by multipath fading, doppler spread, frequency selectivity, and background noise [1], [2]. The mobile unit that is forced to contend with this environment must be kept as small as possible to ensure portability. The portability constraint in turn places a constraint on battery weight, introducing a critical trade-off between transmitter power consumption and battery lifetime. And finally, to further complicate matters, the amount of spectrum available for a given cellular system is quite small. Cellular operators are victims of their own success in the sense that the major cellular telephone markets (Los Angeles, New York, and Chicago) are operating at or beyond capacity. It is estimated that the 20 MHz frequency band allocated by the FCC for use in American cellular telephony will have to accommodate 15 - 20 million users by the end of the century [1]. It follows that a digital system that can put the greatest number of channels into a fixed bandwidth is going to be the system of choice.

The impact of these constraints on system design are fairly straightforward (though occasionally conflicting). We begin with the modulation format. Channel fading indicates the use of constellations in which radial compression of the constellation will not cause a transmitted signal to look like another valid signal at the receiver (e.g., phase modulation, or frequency shift keying). This problem can be circumvented through the use of channel sounding techniques, but this may not be an efficient solution for small mobile transmitters. The power constraints also affect the choice of a modulation format. In order to maximize power efficiency, nonlinear power amplifiers are frequently used in the mobile unit. When a filtered, nonconstant envelope signal passes through a nonlinear device, the side lobes of the signal are restored, and crosstalk and phase jitter are introduced. Adjacent channel and cochannel interference result, and the power efficiency of the receiver is reduced. Constant envelope modulation formats are thus preferred.

Even with a constant envelope signal, frequency selectivity and doppler spread severely limit the performance of spectrally-efficient modulation formats. Interchannel signaling in the Advanced Mobile Phone System (AMPS) achieved a spectral efficiency of 0.33 bits/sec/Hz using FM modulation. The American Digital Cellular standard (ADC) achieves 1.62 bits/sec/Hz [1] using $\pi/4$-QPSK. This is not terribly impressive when compared with a Group 3 FAX machine, which can achieve up to 6 bits/sec/Hz using a 128-signal QAM constellation with trellis coded modulation. The difference between the two lies almost entirely in the relative quality of their channels. In summary, the choice of formats for land mobile radio is quite limited, with $\pi/4$-QPSK [1], Gaussian Minimum Shift Keying (GMSK) [6], or 4-PAM/FM [12] being the current favorites.

The error control system must contend with both varying channel attenuation (fading) and background noise. A powerful random error correcting code that can also cope with burst errors is thus a good choice. The burst error correcting capability of the codes is usually enhanced through interleaving [4], [5]. Interleaving introduces delay into the system, so the degree of interleaving that can be used is a function of the application.

Fading can generally be detected at the receiver, so any error control system that can make use of this "side information" will have an advantage. The Viterbi algorithm provides maximum likelihood soft decision decoding for convolutional and

trellis codes. On an additive white Gaussian noise channel, soft decision decoding provides approximately 3 dB coding gain over an equivalent hard decision decoding system. This gain increases to over 9 dB when the channel is Rayleigh fading [4]. Unfortunately convolutional and trellis codes are highly susceptible to burst errors, and must thus be supported by interleaving when used over a fading channel. Reed-Solomon codes, on the other hand, provide excellent burst error protection as well as significant random error correction [13]. The most popular Reed-Solomon decoding algorithms can also be used to decode erasures and errors simultaneously. But, unfortunately, any form of soft decision decoding beyond erasure decoding is still highly inefficient for all but the smallest of the Reed-Solomon codes.

3. Voice Communication

Voice communication systems have two basic properties that separate them from other forms of communication: they can tolerate an extremely high bit error rate ($\leq 10^{-2}$), but are extremely intolerant of delay (for toll quality transmission the delay must be less than or equal to 0.1 seconds). An uncompressed voice signal of reasonable quality can be transmitted at 36 Kbps. This can be reduced to as low as 2400 bps through the use of advanced compression techniques, but such techniques consume power and also render the signal much more intolerant of channel errors. We will now consider the means by which the European Telecommunications Standards Institute (and a consortium of extremely interested industrial/government partners) coped with these issues in the development of Europe's mobile telephone standard, GSM.

3.1 Example System

The ETSI/GSM modulation format is Gaussian minimum shift keying (GMSK) with differential encoding. Transmissions from the base stations to the mobiles use both FDMA and TDMA on the 935-960 MHz band, while the transmissions from mobile to base are time division multiplexed on the 900-915 MHz band. The mobile to base channels support eight users simultaneously at a cumulative bit rate of 270.83 kbps [7].

Regular pulse excited-long-term prediction (RPE-LTP) is used to reduce the data rate for individual speech signal to 13 Kbps. The RPE-LTP frame consists of 260 bits that require varying levels of error protection. Listening tests were conducted to determine the relative sensitivities of the various bits to error [9]. It was decided to allocate the 260 bits among three classes, with each class receiving a different level of error protection. The resulting coding system is shown in Figure 1. The first fifty bits of the frame are encoded for error detection using a CRC code with generator polynomial $g(x) = x^3 + x + 1$. Four zero bits are then appended to the code word, and the resulting word is encoded using a rate 1/2 convolutional code with transfer function matrix $[1 + D^3 + D^4 \quad 1 + D + D^3 + D^4]$. The next 132 bits in the speech frame are just encoded with the convolutional code, while the remaining 78 bits are left uncoded. The 456-bit coded frame is divided into eight 57-bit blocks. These blocks are then diagonally interleaved over eight time slots, as shown in Figure 2. The blocks of 114 bits that result are transmitted as bursts over the TDMA channel.

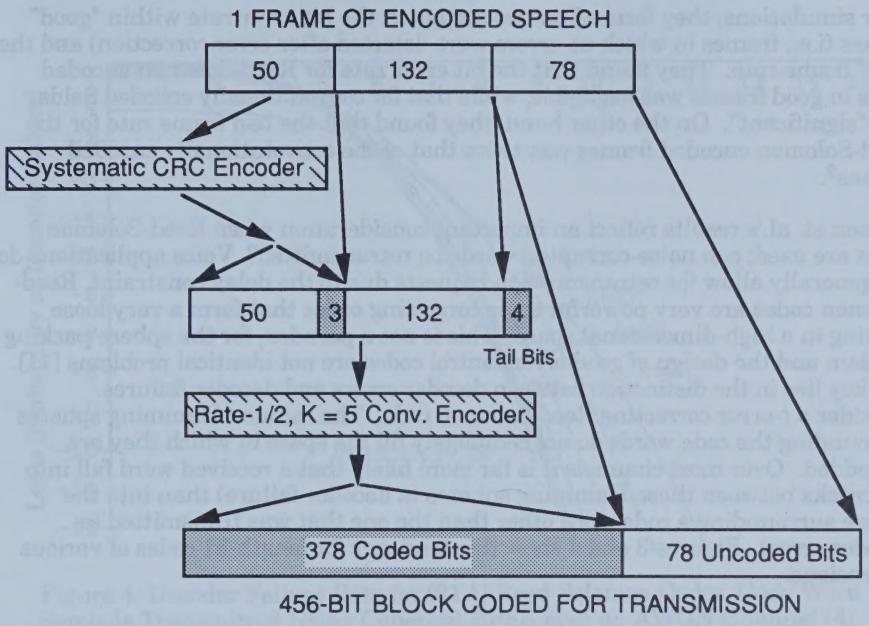


Figure 1: GSM Error Control Coding Format

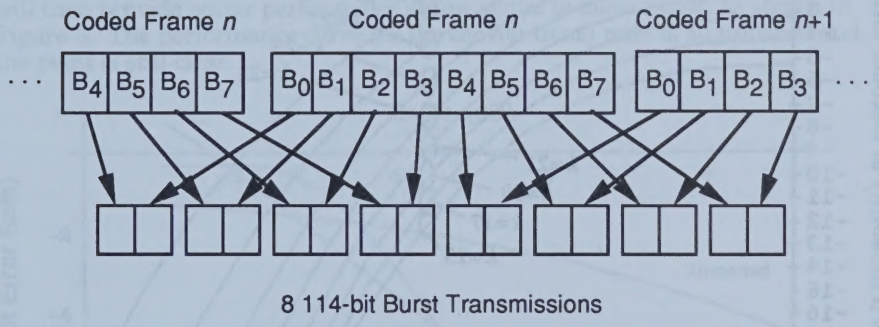


Figure 2: Diagonal Interleaving in GSM [8]

3.2 Comments

Several studies were conducted by a variety of researchers to aid in the selection of the error control system for GSM (e.g., [7], [8], [9], and [10]). These studies typically compared convolutional or punctured convolutional codes to BCH and Reed-Solomon codes. The results were fairly consistent. Nielsen et. al. began by noting that an error control system based on punctured convolutional codes could more easily provide the variable error protection required by the speech compression algorithm than a comparable Reed-Solomon-based system [9]. In

their simulations, they focused on two statistics: the bit error rate within "good" frames (i.e., frames in which no errors were detected after error correction) and the "bad" frame rate. They found that the bit error rate for Reed-Solomon encoded fields in good frames was negligible, while that for convolutionally encoded fields was "significant". On the other hand, they found that the bad frame rate for the Reed-Solomon encoded frames was twice that of the convolutionally encoded frames[2].

Nielsen et. al.'s results reflect an important consideration when Reed-Solomon codes are used: can noise-corrupted words be retransmitted? Voice applications do not generally allow for retransmission requests due to the delay constraint. Reed-Solomon codes are very powerful error correcting codes that form a very loose packing in a high-dimensional space. This is not a paradox, for the sphere packing problem and the design of good error control codes are not identical problems [11]. The key lies in the distinction between decoder errors and decoder failures. Consider a t-error correcting Reed-Solomon code. The radius-t Hamming spheres surrounding the code words do not completely fill the space in which they are embedded. Over most channels it is far more likely that a received word fall into the cracks between these Hamming spheres (a decoder failure) than into the sphere surrounding a code word other than the one that was transmitted (a decoder error). Figures 3 and 4 show this tendency for length-31 codes of various dimensions.

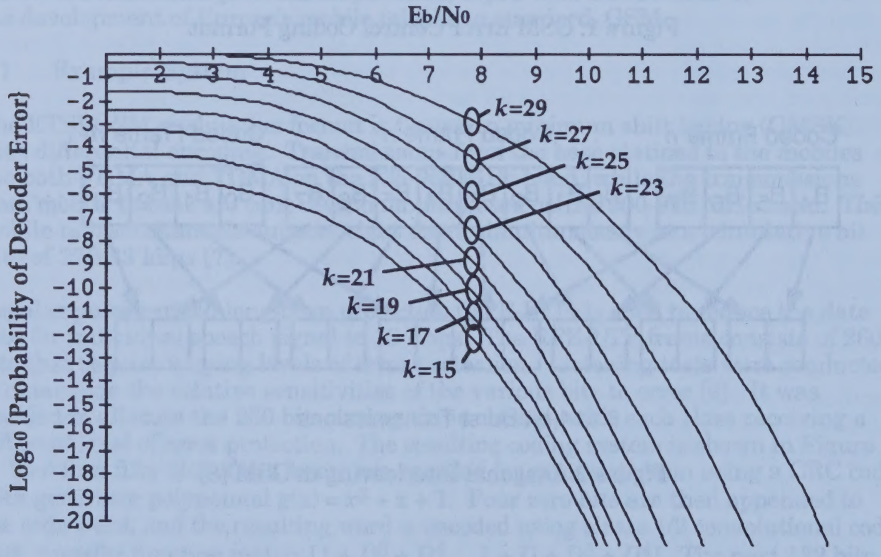


Figure 3: Decoder Error Rate for (31,k) Reed-Solomon Codes, Code Word Symbols Transmitted Using Coherent BPSK over an AWGN Channel [4].

[2]Nielson et. al. used a multistage RS decoding algorithm that used erasure decoding to reduce the bad frame rate; without it, the disparity in the bad frame rate between the RS and convolutionally encoded systems would have been even greater.

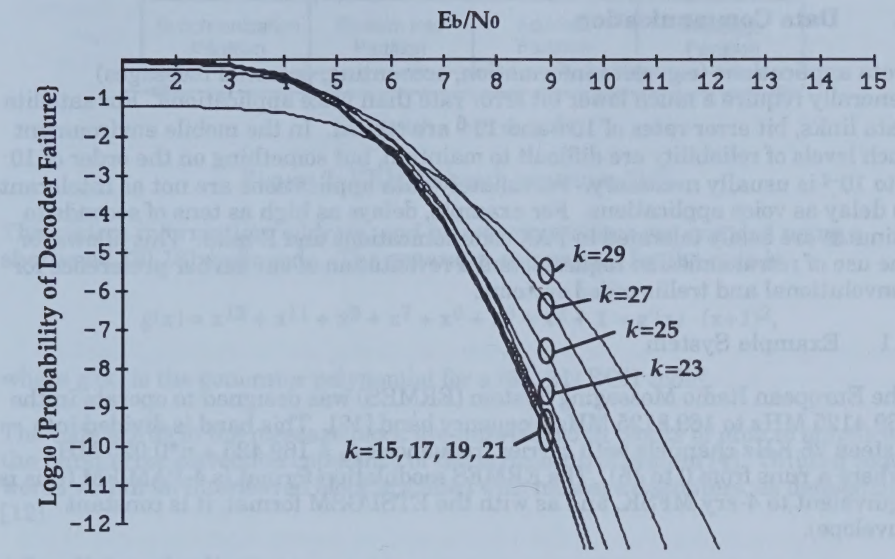


Figure 4: Decoder Failure Rate for (31,k) Reed-Solomon Codes, Code Word Symbols Transmitted Using Coherent BPSK over an AWGN Channel [4].

Unfortunately, in the FEC mode we must equate decoder failures with decoder errors (or bad frames). The FEC Viterbi-decoded convolutional or TCM system will then provide better performance at low signal to noise ratios, as shown in Figure 5. The performance curve for the convolutional code is an upper bound, but the point is still clear.

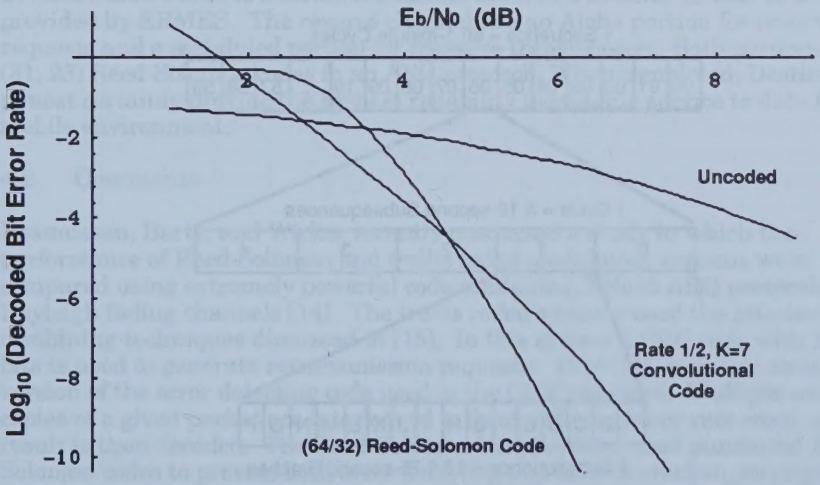


Figure 5: Comparison of the performance of a convolutional and a Reed-Solomon code over an AWGN channel [4].

4. Data Communication

Data applications (e.g. sales information, accounting, personal messages) generally require a much lower bit error rate than voice applications. For satellite data links, bit error rates of 10^{-5} and 10^{-6} are typical. In the mobile environment such levels of reliability are difficult to maintain, but something on the order of 10^{-3} to 10^{-4} is usually necessary. Fortunately data applications are not as intolerant to delay as voice applications. For example, delays as high as tens of seconds to minutes are easily tolerated in FAX communications and E-mail. This allows for the use of retransmission requests, and a revisitation of our earlier preference for convolutional and trellis coded systems.

4.1 Example System

The European Radio Messaging System (ERMES) was designed to operate in the 169.4125 MHz to 169.8125 MHz frequency band [12]. This band is divided into sixteen 25 KHz channels with carrier frequencies $f_n = 169.425 + n*0.025$ MHz (where n runs from 0 to 15). The ERMES modulation format is 4-PAM/FM (this is equivalent to 4-ary MFSK, and as with the ETSI/GSM format, it is constant envelope).

Each ERMES channel is divided in time into a series of 1-hour sequences, as shown in Figure 6. Each sequence is then divided into sixty 1-minute cycles, which are in turn divided into five 12-second subsequences. The subsequences are divided into batches, which are the basic units of the protocol. The batches are denoted by letters (A through P), and are divided into synchronization, system information, address, and message partitions, as shown in Figure 7. Each receiving unit is assigned to a single batch type, and can only be addressed within that particular type. The batches are arranged within the sixteen channels so that a given receiver capable of scanning all sixteen channels need never miss its designated batch in any channel.

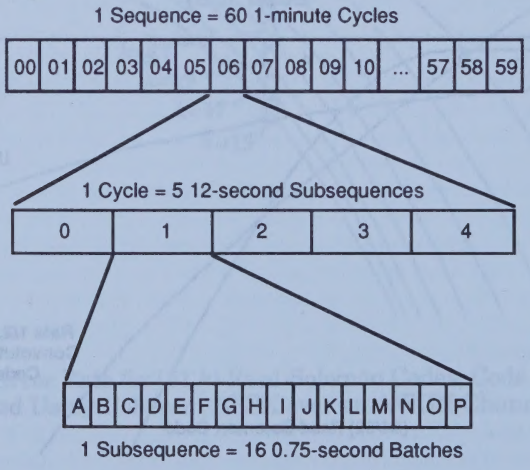

1 Sequence = 60 1-minute Cycles
00 01 02 03 04 05 06 07 08 09 10 ... 57 58 59
1 Cycle = 5 12-second Subsequences
0 1 2 3 4
A B C D E F G H I J K L M N O P
1 Subsequence = 16 0.75-second Batches

Figure 6: ERMES radio protocol for channel 1 [12].

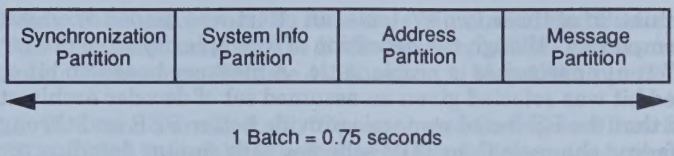


Figure 7: ERMES batch structure [12].

The system information, address, and message partitions are encoded using a shortened (30,18) cyclic code. The generator polynomial for the code is

$$g(x) = x^{12} + x^{11} + x^9 + x^7 + x^6 + x^3 + x^2 + 1 = g'(x) \cdot (x+1)^2,$$

where $g'(x)$ is the generator polynomial for a (31, 21) BCH code[3].

The code words in the message block are interleaved in blocks of nine to enhance the burst error correction capability of the error control system [4]. Unused code words within an interleaved block are filled with message delimiters, as defined in [12].

4.2 Future Applications

Mobile Telecommunication Technologies (MTEL) was granted a Pioneer's Preference by the FCC to develop and deploy Destineer, a nationwide wireless messaging service. The Pioneer's Preference was recently strengthened by MTEL's acquisition of several channels in the 900 - 1000 MHz band. Destineer will use these channels to provide various two-way messaging services to mobile units. It will be possible for users to access the system through their mobile units, as well as DTMF telephones, E-mail, computer modems, INTERNET, FAX, and IVR. The forward channel in the air interface will use interleaved blocks of (31, 21) BCH code words to address the mobile units in a manner similar to that provided by ERMES. The reverse channel has an Aloha portion for reservation requests and a scheduled portion for message transmission. Both portions will use (31, 23) Reed-Solomon codes in an ARQ protocol. When completed, Destineer will almost certainly provide the highest reliability messaging service to date for the mobile environment.

4.2 Comments

Rasmussen, Bartz, and Wicker recently conducted a study in which the performance of Reed-Solomon and trellis coded modulation systems were compared using extremely powerful code combining, hybrid-ARQ protocols over Rayleigh fading channels [14]. The trellis coded systems used the interleaved code combining techniques discussed in [15]. In this system a CRC code with 16 parity bits is used to generate retransmission requests. This CRC code is a stronger version of the error detecting code used in the GSM standard. Multiple received copies of a given packet are interleaved to form a single, lower rate word, and the result is then decoded. The Reed-Solomon-based system used punctured Reed-Solomon codes to provide both error detection and error correction, as originally discussed in [13] and [16]. In this system, multiple received words are concatenated to form noise corrupted words from low-rate Reed-Solomon codes.

---

[3]This (31,21) BCH code is also used in POCSAG, the current digital paging standard in the U.S.

In the evaluation of these two systems, an effort was made to compare systems of equal complexity - though the definition of a complexity measure for these rather different approaches is problematic. A measure based on bit operations per decoded bit was selected given an assumed set of decoder architectures. It was found that the RS-based protocols provide better BER and throughput over Rayleigh fading channels than TCM schemes with similar decoding complexity. The performance of the TCM schemes, however, is far less sensitive to a decrease in the allowed complexity. The reason for the better performance by the RS-based protocol is the inherently high residual error-detection capability of RS codes, as discussed in Section 3.2.

5. Conclusions

In this paper we have considered the constraints imposed by the mobile radio channel on voice and data applications. These include limited power, limited bandwidth, multipath fading, and in some cases frequency selectivity. Recently developed voice and data systems, GSM and ERMES, were considered in some detail to illustrate the design process and to underscore the rationale behind the selection of certain error control systems over others. It was shown that the intolerance to delay shown by voice applications leads to a preference for soft-decision convolutionally encoded or trellis coded error control systems. When the delay constraint is removed, the preference shifts to block coded systems, with a particular emphasis on Reed-Solomon codes. In all cases, the fading channel forces the designer to rely on constant envelope modulation formats and interleaving to enhance the performance of the error control system.

References

[1] Feher, K., "MODEMS for Emerging Digital Cellular-Mobile Radio System," *IEEE Transactions on Vehicular Technology*, Volume 40, Number 2, May 1991, pp. 355 - 365.

[2] Lee, W. C. Y., *Mobile Communications Design Fundamentals*, Indianapolis: Howard W. Sams, 1986.

[3] Wicker, S. B., "Reed-Solomon Error Control Coding for Data Transmission over Rayleigh Fading Channels with Feedback," *IEEE Transactions on Vehicular Technology*, Volume 41, Number 2, pp. 124 - 133, May 1992.

[4] Wicker, S. B., *Error Control Systems for Digital Communication and Storage*, Englewood Cliffs: Prentice Hall, 1994.

[5] Wicker, S. B., "High Reliability Data Transfer over the Land Mobile Radio Channel," *IEEE Transactions on Vehicular Technology*, Volume 39, No 1, pp. 48 - 55, February 1990.

[6] K. Hirade, "Mobile Radio Communications," in *Advanced Digital Communications*, Chapter 10, edited by K. Feher, Englewood Cliffs: Prentice Hall, 1987.

[7] G. D'Aria, F. Muratore, and V. Palestini, "Simulation and Performance of the Pan-European Land Mobile Radio System," *IEEE Transactions on Vehicular Technology*, Volume 41, Number 2, May 1992, pp. 177 - 189.

[8] Lee, L. H. C. and Farrell, P. G., "Concatenated Coding and Interleaving for Half-Rate GSM Digital Mobile Radio System," *Electronics Letters*, Volume 27, Number 14, July 4, 1991, pp. 1274-1276.

[9] Nielsen, H., Mikkelson, K. B., Hansen, H. B., Wu, Y., Larsen, K. J., Sørenson, J. Å, "Comparative Study of Error Correction Coding Schemes for the Half-Rate GSM Channel," *Proceedings of ICASSP-92*, pp. II-129 - II-132.

[10] Atungsiri, S. A., Sweeney, P., Soheili, R., Kondoz, A. M., and Evans, B. G., "Soft RS Codes for Half-Rate GSM Channel," *Electronics Letters*, Volume 27, Number 2, January 17, 1991, pp. 176-178.

[11] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices, and Groups*, New York: Springer-Verlag, 1988.

[12] *European Radio Messaging System (ERMES), Part 4: Air Interface Specification*, European Telecommunications Standards Institute.

[13] Wicker, S. B. and Bhargava, V. K. (editors), *Reed-Solomon Codes and Their Applications*, Piscataway: IEEE Press, 1994.

[14] Rasmussen, L. K., Bartz, M. J., and Wicker, S. B., "A Comparison of Trellis Coded and Reed-Solomon Coded Hybrid-ARQ Protocols over Slowly Fading Rayleigh Channels," submitted to the *Wireless Personal Communications*, August 1, 1994.

[15] Rasmussen, L. and Wicker, S. B., "Trellis Coded Hybrid-ARQ Protocols over AWGN and Slowly Fading Rician Channels," *IEEE Transactions on Information Theory*, March, 1994.

[16] Wicker, S. B. and Bartz, M., "Type-II Hybrid-ARQ Protocols Using Punctured MDS Codes," *IEEE Transactions on Communications*, Vol. 42, No. 4, pp. 1431-1440, April 1994.

CODING ALTERNATIVES FOR DIRECT-SEQUENCE SPREAD-SPECTRUM MULTIPLE-ACCESS COMMUNICATIONS

Wayne E. Stark
University of Michigan
Ann Arbor, MI

and

James S. Lehnert
Purdue University
West Lafayette, IN

Abstract

In this paper we examine the performance of different coding schemes for spread-spectrum communication with multiple-access interference. We consider nonlinear codes that have large distance for a given rate and we examine trellis-coded versions of these nonlinear codes (also called finite state codes). We develop a bound on the error probability for these trellis codes on the additive white Gaussian noise channel that is tighter than the standard union bound. We also consider the improved Gaussian approximation [1] when the channel has multiple-access interference. Our results indicate that the standard union bound and standard Gaussian approximation can be significantly different from the more accurate results obtained from the improved Gaussian approximation or the improved union bound.

1 Introduction

In a conventional direct-sequence spread-spectrum (DS-SS) communication system a single data bit is transmitted using a pseudo-random sequence or its negative and binary phase shift keying (BPSK). Typically, there are N chips of the pseudo-random sequence for each information bit transmitted. The number of chips per bit is a measure of the processing gain of the system when used in an environment with multiple-access interference or multipath fading. The demodulation consists of (assuming perfect synchronization) a correlator matched to the pseudo-random sequence followed by a decision device. In addition an error-correcting code such as a convolutional code or block code can be used to provide additional protection, usually at the expense of data rate.

An alternative description of this system, which makes generalizations and modifications easier to visualize, is as a stream of information bits that are encoded by a repetition code of length N. Each bit is repeated N times. The output of the encoder is then scrambled by a pseudo-random sequence before being modulated using BPSK. The receiver consists of a demodulator (for each chip or code symbol) followed by a decoder, which is a soft decision decoder for the repetition code. If an error correcting code is used in addition, then essentially the system employs concatenated coding.

Viewing a DS-SS system in this manner makes it clear that one can use other codes besides the repetition code, which while having good distance properties has poor rate. In [2], orthogonal codes were considered. Orthogonal codes have half the distance of repetition codes, but have N times as many signals, so the data rate can be increased by up to a factor of $\log_2(N)$. In the IS-95 system for digital cellular, orthogonal codes are used on the reverse link (mobile-to-base). Biorthogonal codes have the same minimum distance as orthogonal codes, but can transmit one more bit of information. The rate is a factor of $\log_2(N) + 1$ larger than the repetition codes. In [3], it was shown that by using a trellis code defined over a biorthogonal set, a gain in distance and rate is possible compared to conventional DS-SS systems.

While distance and rate are important parameters of a code and determine the performance in additive white Gaussian noise for high signal-to-noise ratios, it is also important to consider the number of nearest neighbor codewords. This is especially true when operating at low signal-to-noise ratios. For the repetition code there is just one nearest neighbor codeword. For the orthogonal codes there are $\log_2(N) - 1$ nearest neighbors. Clearly, one can use a subcode of the orthogonal code in order to reduce the number of nearest neighbors. This, however, sacrifices code rate. Another method of reducing the number of nearest neighbors is to use a combination of an orthogonal code with a trellis. The number of neighbors at the minimum distance can be reduced without sacrificing rate at the expense of complexity.

In this paper we wish to explore alternative coding systems when used in a DS-SS system. Our goal is to achieve higher data rate and lower error probability for the same number of interfering users. The codes we consider include nonlinear codes such as the Nordstrom-Robinson (NR) code and the Kerdock codes [4]. These codes have better distance/rate properties than linear codes and have efficient soft-decision decoding algorithms. The distance profiles of these codes can be slightly improved by employing a trellis [5]. This also reduces by a considerable amount the number of nearest neighbors.

The remainder of the paper is organized as follows. In the next section we discuss these coding strategies and the standard union bound. We show how the union bound can be improved for the trellis coded case. Then, in Section 3, we discuss the multiple-access interference and derive approximate bounds based on the approximation of the interference as Gaussian. In Section 4, we show how to improve the accuracy of the bounds with the improved Gaussian approximation [1]. Finally, we present some numerical results in Section 5.

2 Codes

Consider a standard, but not necessarily linear, binary error correcting code C. The code has a length n and a size, or number of codewords, M. That is, the code consists of M vectors of length n from the alphabet $\{0, 1\}$. For all codes considered in this paper $M = 2^k$ for some k. The weight enumerator for the code; denoted by A_i, $i = 0, 2, ..., n$; is the number of codewords of (Hamming) weight i. The distance distribution $W_i(x)$ is the number of codewords of distance i from codeword x. For linear codes the distance distribution and the weight distribution are identical. This is also true for the codes considered in this paper, although the codes will not all be linear. This property is sometimes called geometric uniformity or distance uniformity. When a code has this property, the analysis and simulation of the performance is greatly simplified because the conditional error probability given any particular codeword is transmitted does not depend on which codeword is transmitted. Thus, we can assume any codeword is transmitted and get the unconditional error probability. Finally, when transmitting a binary vector with components $\{0, 1\}$ the translation to the components in the set $\{\pm 1\}$ via $s_i = (-1)^{x_i}\sqrt{E}$ will take into account the usual BPSK modulation.

One class of interesting codes is the biorthogonal code. This code has $M = 2n$. The code is formed by first finding any set of orthogonal basis vectors for the space of binary n vectors. The code consists of these vectors and their complements. One convenient set

of orthogonal vectors is the Hadamard set constructed recursively via

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad H_{2^m} = \begin{bmatrix} H_{2^{m-1}} & H_{2^{m-1}} \\ H_{2^{m-1}} & -H_{2^{m-1}} \end{bmatrix}$$

The orthogonal vectors are the rows of the Hadamard matrix. The code which consists of these vectors and their negatives is a biorthogonal code. The distance distribution is

$$A_i = \begin{cases} 1 & i = 0 \\ M-2 & i = n/2 \\ 1 & i = n \end{cases}$$

This code transmits $\log_2(M)$ bits of information using the channel n times.

If we start with the biorthogonal code, we can increase the number of codewords by adding selected cosets of the biorthogonal code to the original code. When we do this, we slightly decrease the minimum distance of the code. However, the decoding algorithm is only modified slightly. The algorithm to decode the original code can be used to decode any of the cosets. If the received vector is first multiplied (component by component) with the coset leader for a particular coset, then the decoding algorithm for the original code can be applied to the modified received vector. Once all the cosets are decoded separately, an overall decision is made by choosing the most likely vector from the codewords decoded for each coset. For the biorthogonal code of length 16 with 32 codewords and minimum distance 8, we can increase the number of codewords to 64 with the minimum distance decreasing to 6. In fact, if we continue to add cosets we can increase the number of codewords up to 256 with the minimum distance staying at 6. When we do this, we get the Nordstrom-Robinson nonlinear code. Similarly, when we start with a biorthogonal code of length 64 with 128 codewords (7 information bits) and minimum distance 32, we can increase the size of the code to 2048 and decrease the distance to 28. This is one of the Kerdock codes. As a final example, the biorthogonal code of length 256 with 512 codewords of distance 128 can be increased in size and rate to have $65356 = 2^{16}$ codewords with minimum distance decreasing from 132 to 120.

The performance of these codes on an AWGN channel can be easily bounded with the union bound. Since these codes are all geometrically uniform, the union bound on codeword error probability is given by

$$Q(\sqrt{\frac{2Ed_{min}}{N_0}}) \leq P_e \leq \sum_{i=d_{min}}^{n} A_i Q(\sqrt{\frac{2Ei}{N_0}})$$

where E is the energy transmitted per chip. This is related to E_b, the energy per information bit, via

$$E_b = nE/\log_2(M).$$

The weight enumerator for the Kerdock codes is shown below for $n = 16, 64$, and 256. The $n = 16$ code is the Nordstrom-Robinson code.

$$n = 16: \quad A_i = \begin{cases} 1 & i = 0 \\ 112 & i = 6 \\ 30 & i = 8 \\ 112 & i = 10 \\ 1 & i = 16 \end{cases}, \quad n = 64: \quad A_i = \begin{cases} 1 & i = 0 \\ 1984 & i = 28 \\ 126 & i = 32 \\ 1984 & i = 36 \\ 1 & i = 64 \end{cases}, \quad n = 256: \quad A_i = \begin{cases} 1 & i = 0 \\ 32512 & i = 120 \\ 510 & i = 128 \\ 32512 & i = 136 \\ 1 & i = 256 \end{cases}$$

If we consider the $n = 16$ code which has $256 = 2^8$ codewords, the rate is 8/16, and the minimum distance is 6. The biorthogonal code of length 16 has rate 5/16 and minimum distance 8. Thus we have decreased the minimum distance by 25%, but have increase the rate by 80%. The complexity of the decoding of the $n = 16$ Nordstrom-Robinson code is roughly 8 times that of the biorthogonal code. If additional complexity is tolerable, then we can modify the NR code to get back the loss in distance without changing the rate. The method is to employ a trellis code over the NR code. Consider an 8-state trellis which is fully connected. Partition the NR code into 8 sets, one set being the biorthogonal code and the other seven sets being cosets of this code. If we label the outgoing branches of the trellis with these sets, we form a trellis code. To encode, we divide the 8 bits of information into 5 bits and 3 bits. The 3 bits determine which coset, and hence which outgoing branch, of the trellis, while the 5 bits determine which codeword in that set. If we label the sets $B_0, ..., B_7$ appropriately, we can change the distance structure of the code. Consider the top state in the trellis. The outgoing branches are labeled consecutively starting with the branch that stays in the top state (labeled with B_0) and proceeding to the branch that starts in the top state and goes to the bottom state (labeled with B_7). The next state is labeled in a similar manner except that the branch going to the top state is labeled B_7, the next state is labeled B_0, and so on. The minimum distance of the new code is 8 and the number of nearest neighbors is reduced back to 30. The weight enumerator of the new code can be easily shown to be

$$A(D) = H(D) + \frac{G^2(D)}{1 - G(D)}$$

where $H(D) = 1 + 30D^8 + D^{16}$ is the weight enumerator of the biorthogonal code and $G(D) = 16D^6 + 16D^{10}$ is the weight enumerator of the coset of the biorthogonal relative to a codeword in the biorthogonal code. Note that we have simultaneously increased the minimum distance of the code and decreased the number of nearest neighbors without changing the code rate. Furthermore, the code also has the geometric uniformity property, which makes it easy to analyze or simulate. The disadvantage is increased complexity of decoding. However, an 8-state code is relatively low in complexity. This is an example of finite state codes described in [5].

Another example from [5] is a trellis over the family of Kerdock codes. The Kerdock code of length 64 would have 32 states, while the Kerdock code of length 256 would have 128 states.

The analysis of performance of these codes can be tightened from the union bound by recognizing that the codes consists of biorthogonal codes and their cosets. Since the error probability of the biorthogonal code can be expressed in terms of a single integral, we can improve the union bound on the error probability. This can be done for either the standard Kerdock code or when the code is combined with a trellis. The improvement should be more for the case when the code is on the trellis because the minimum distance error events are codewords within the biorthogonal code. First, we state the standard union bound for additive white Gaussian noise and an upper bound to the union bound obtained by calculating the exact pairwise error probability for the first L terms and bounding the remaining terms by the Bhattacharyya bound, that is

$$P_e \leq \sum_{i=d_{min}}^{\infty} A_i Q\left(\sqrt{\frac{2Ei}{N_0}}\right) \leq \sum_{i=d_{min}}^{L} A_i \left[D^i - Q\left(\sqrt{\frac{2Ei}{N_0}}\right)\right] + A(e^{-E/N_0})$$

where $A(x) = \sum A_i x^i$. For large L, the upper bound to the union bound becomes arbitrarily tight.

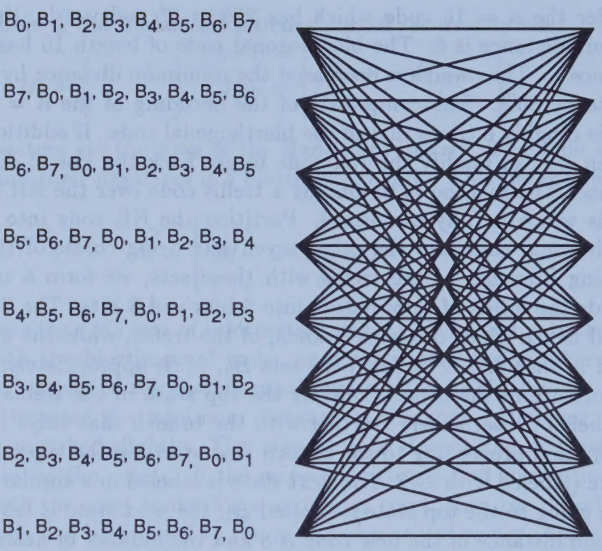


Figure 1: Trellis Coded Nordstrom-Robinson Code. Labels for branches are cosets of the biorthogonal code proceeding from the top branch to the bottom as the labels go from left to right.

Now consider the trellis coded Nordstrom-Robinson (TCNR) code. We can get a very good approximation to the union bound by taking the first four terms exactly and then upper bounding the remainder.

$$P_e \leq 30(D^8 - P_2(8)) + 256(D^{12} - P_2(12)) + 513(D^{16} - P_2(16)) + 12888(D^{22} - P_2(22)) + H(D) + \frac{G^2(D)}{1 - G(D)}$$

where $D = e^{-E/N_0}$.

Now we show how to tighten the above upper bound to the union bound. As before, B_0 is the biorthogonal subcode and $B_1, ..., B_7$ are the cosets of the biorthogonal code. The union of $B_0, ..., B_7$ make up the NR code. Because of the geometric uniformity of this code, we can assume without loss of generality that the all zeros codeword is transmitted. Let $c_{i,j}$ be the j-th codeword in B_i. We let $c_{0,0}$ be the all zero codeword. The error probability is then the probability that we decode to a codeword other than $c_{0,0}$. Let $\hat{c}$ denote the decoder output and $\tilde{B}_0$ denote $B_0 - c_{0,0}$. Then

$$\begin{aligned} P_e &= P\left[\hat{c} \in \tilde{B}_0 \cup B_1 \cup \cdots \cup B_7\right] \\ &\leq P\left[\hat{c} \in \tilde{B}_0\right] + P\left[\hat{c} \in B_1 \cup \cdots \cup B_7\right] \end{aligned}$$

Now it is clear that the first term is just the error probability for a biorthogonal code. The last term can be bounded by the standard union bound. Because the minimum distance event is a decoding error not to the biorthogonal code but to a codeword in any one of the cosets, it is unlikely that this will improve the union bound considerably for the standard Kerdock code. However, if we consider the trellis coded Kerdock code,

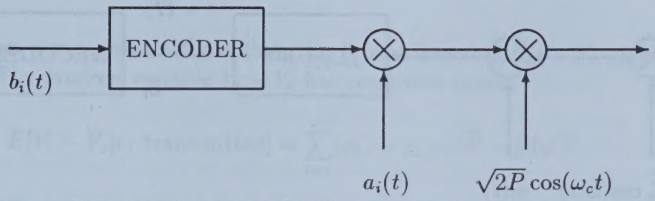


Figure 2: Model of the Transmitter for User i.

then the minimum distance error events are the codewords in the biorthogonal code. Thus, the improvement in the union bound should be more significant. Let $F(d) = G^2(D)/(1 - G(D)) = \sum_{l=d_1}^{\infty} f_i D^l$. Then the improved union bound for the trellis coded NR code becomes

$$P_e \leq P_{32} + \sum_{l=d_1}^{\infty} f_i D^i$$

where P_{32} is the exact error probability for a biorthogonal signal set of size 16 and D^i is the Bhattacharyya bound on the error probability between two codewords that differ in i places. This improved union bound applies to the trellis coded Kerdock codes as well. We can get a further improvement by determining the pairwise error probability exactly for the first L terms in the sum but the expected improvement is minimal since the dominant terms are the minimum distance terms.

3 Multiple-Access Interference

In this section we derive approximations to the codeword error probabilities for these codes. We assume random sequences for the different users and approximate the interference by Gaussian interference, or conditionally Gaussian interference when given the delays and phases of the interfering signals. We first introduce the system model and then analyze the performance with the above mentioned approximations.

The model of the system is shown below. There are K users that encode a data stream as described above. The output of the encoder is then multiplied by a pseudo-random sequence unique to each user. The receiver multiplies the incoming signal by a carrier reference signal and its assigned spreading sequence before processing the product with a filter matched to the chip (code symbol) waveform. The resulting sequence is the input to the decoder, which finds the closest codeword to the input sequence.

The input to the encoder is a sequence of information bits

$$b_i(t) = \sum_{l=-\infty}^{\infty} b_{i,l} p_T(t - lT)$$

where $p_T(t)$ is the standard unit amplitude pulse of duration T starting at time 0. The spreading sequence has the same form except it has pulses of duration T_c:

$$a_i(t) = \sum_{l=-\infty}^{\infty} a_{i,l} p_{T_c}(t - lT_c).$$

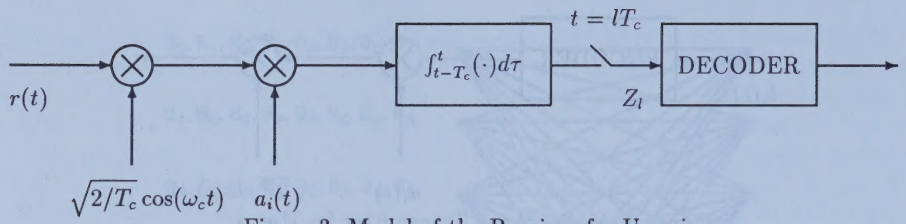


Figure 3: Model of the Receiver for User i.

The output of the encoder $c(t)$ is also a sequence of rectangular pulses. Namely,

$$c_i(t) = \sum_{l=-\infty}^{\infty} c_{i,l} p_{T_c}(t - lT_c).$$

The encoder processes k information bits (which span kT seconds) and produces n channel bits which span the same amount of time with each channel bits lasting T_c seconds. Thus, $nT_c = kT$ or $k/n = T_c/T$. The information bits $b_{i,0}, ..., b_{i,k-1}$ are mapped into channel symbols $c_{i,0}, ..., c_{i,n-1}$. This repeats in a block structure. For the conventional direct-sequence system $k = 1$ and n is the number of chips per bit.

Thus, the transmitted signal for user i is

$$s_i(t) = \sqrt{2P} a_i(t) c_i(t) \cos(\omega_c t).$$

For this paper the channel is a standard multiple-access channel with the channel output being the sum of the K transmitted signals (with each signal randomly delayed) and additive white Gaussian noise with power spectral density $N_0/2$. Thus

$$r(t) = \sum_{k=1}^{K} s_k(t - \tau_k) + n(t)$$

where $n(t)$ is the noise. We assume that all users have the same power. The receiver for user i is shown in Figure 2.

Let the output at time lT_c of the filter matched to a chip pulse be denoted by Z_l. If there are K users with different spreading sequences, then the output of the filter matched to a single chip (code symbol) is

$$Z_l = \sqrt{E} x_l + \eta_l + I_l$$

where η_l is a zero mean Gaussian random variable with variance $N_0/2$ and I_l is the multiple-access interference. The decoder correlates the received sequence Z_l with each of the possible codewords to determine which one has the largest correlation. In determining the error probability it is important to consider a pair of codewords that differ in only d places. Consider two such codewords c_1 and c_2. The correlation with these two codewords is

$$Y_1 = \sum_{l=1}^{n} Z_l c_{1,l}$$

$$Y_2 = \sum_{l=1}^{n} Z_l c_{2,l}$$

The pairwise error probability is then the probability that $Y_2 > Y_1$ given that x_1 is transmitted. The random variable $Y_1 - Y_2$ has condition mean

$$E[Y_1 - Y_2 | c_1 \text{ transmitted}] = \sum_{l=1}^{n} (c_{1,l} - c_{2,l})\sqrt{E} = 2d\sqrt{E}$$

where $E = PT_c$. The conditional variance is given (assuming uniformly distributed delays, and independent and random sequences) by

$$\text{Var}[Y_1 - Y_2 | c_1 \text{ transmitted}] = 2N_0 d + 4Ed(K-1)/3$$

If the total noise and interference is approximated as Gaussian, then the pairwise error probability between two codewords at Hamming distance d is approximated as

$$P_2(d) \approx Q\left(\sqrt{\frac{Ed}{(K-1)E/3 + N_0/2}}\right).$$

If the two codewords are from a repetition code, then this reduces to the usual error probability and the union bound is tight. The union bound with a Gaussian approximation and ignoring the contribution of thermal noise, i.e. $N_0 = 0$, is given by

$$P_e \leq \sum_{d=d_{min}}^{n} A_d P_2(d) = \sum_{d=d_{min}}^{n} A_d Q\left(\sqrt{\frac{3d}{K-1}}\right).$$

4 Improved Gaussian Approximation

In past work [1], it has been shown that for a standard BPSK DS-SS system with random sequences the Standard Gaussian Approximation (SGA) is overly optimistic for small numbers of interferers. A method for applying a conditional Gaussian approximation given the delays and phases of each interferer, called the Improved Gaussian Approximation (IGA), has been suggested. In this approach, instead of averaging over the entire set of delays and phases of all interferers, one simply averages over the distribution of the variance of the total interference. The reasoning is that the mean and (conditional) variance characterize the multiple-access interference (MAI) when a Gaussian approximation is applied. The distribution of the MAI variance is easily computed, and as the number of spread-spectrum chips per bit grows, the approximation converges to the exact result. In practice, even for relatively small processing gains and small numbers of interferers, the IGA gives good results.

We use a similar approach here for evaluating the pairwise error probability for various codes. However, instead of achieving a component from the transmitted signal that is always N as in [1], the useful component becomes d, the number of places where a pair of codewords differ. (Notice that d=N for a repetition code of length N.)

5 Numerical Results

In Fig.4, the improved union bound is shown for the case of the TCNR code of length 16 concatenated with an inner repetition code of length 8. Calculations are performed using the SGA.

In Fig. 5, different coding schemes with rate 1/16 are compared using the IGA by showing upper and lower bounds on performance. The first one is the standard direct-sequence scheme that employs a repetition code of length 16 to send 16 bits using 256 chips. Another coding scheme uses the Kerdock code of length 256 with 2^{16} codewords to send 16 bits using 256 chips. Although both codes have identical rates, the repetition code has distance 16 (with only 16 nearest neighbors), while the Kerdock code has distance 120 with 32512 nearest neighbors. Clearly, the Kerdock code has higher complexity. The other coding scheme compared is a standard rate 1/2 convolutional code with constraint length 7 (memory 6) concatenated with a rate 1/8 inner repetition code. For the case of the Kerdock code, the upper and lower bounds computed with the SGA are also shown in Fig. 5 with dotted lines. Performance improves dramatically with increases in decoder complexity.

Acknowledgment This research was supported in part by NSF under grant number NCR-9115969.

References

[1] R. K. Morrow and J. S. Lehnert, "Packet throughput in slotted ALOHA DS/SSMA radio systems with random signature sequences", *IEEE Transactions on Communications*, vol. COM-40, pp. 1223–1230, July 1992.

[2] Per K. Enge and Dilip V. Sarwate, "Spread-spectrum multiple-access performance of orthogonal codes: Linear receivers", *IEEE Transactions on Communications*, vol. COM-35, pp. 1309–1319, December 1987.

[3] Wayne E. Stark and Brian D. Woerner, "The application of trellis coding techniques to direct-sequence spread-spectrum multiple-access communications", *Twenty-sixth Annual Allerton Conference on Communication, Control, and Computing*, vol. 26, pp. 1070–1079, October 1988.

[4] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland, Amsterdam, 1978.

[5] Fabrisio Pollara, Robert J. McEliece, and Khaled Abdel-Ghaffer, "Finite state codes", *IEEE Transactions on Information Theory*, vol. 34, pp. 1083–1089, September 1988.

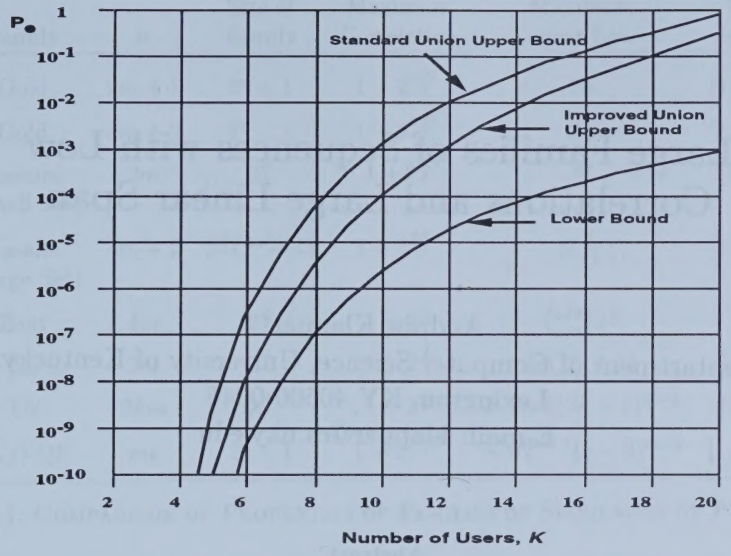


Figure 4: Bounds on Error Probability for the Trellis Coded Nordstrom-Robinson Code.

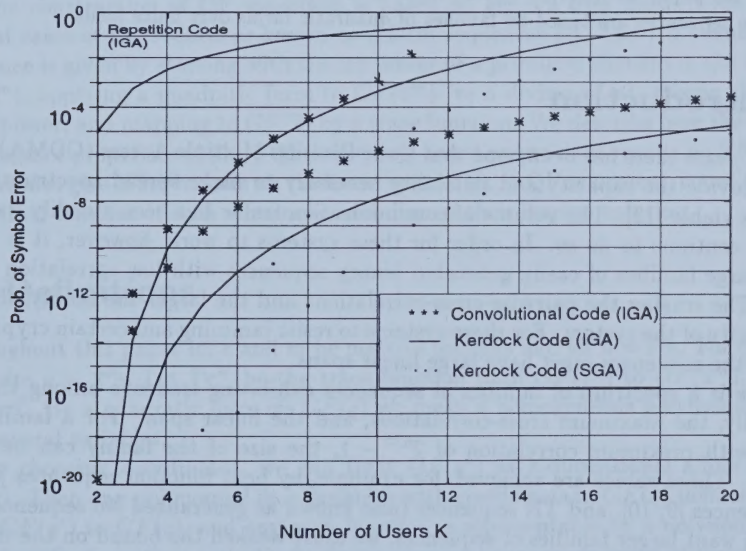


Figure 5: Bounds on Error Probability for the Length 256 Kerdock Code, the Rate 1/2 Convolutional Code with an Inner Repetition Code of Length 8, and the Repetition Code of Length 16.

Large Families of Sequences with Low Correlations and Large Linear Span

Andrew Klapper*
Deptartment of Computer Science, University of Kentucky,
Lexington, KY 40506-0046
E-mail: klapper@cs.uky.edu

Abstract

Communication systems based on the CDMA paradigm, require large families of binary sequences with low pairwise correlation values. Resistance to jamming and certain cryptanalytic attacks requires sequences with large linear span as well. In this paper we describe certain families of sequences that have these properties. The sequences are based on families of quadratic forms over finite fields.

1 Introduction

In recent years there has been hope that Code Division Multiple Access (CDMA) systems would provide the capacity and reliability necessary to make spread spectrum communications viable [12]. The volume of communication traffic has been steadily increasing, and will continue to do so. In order for these systems to work, however, it is essential to find large families of easily generated binary sequences with low correlation function values. The smaller the pairwise cross-correlations and the larger the family, the higher the capacity of the system. For these systems to resist jamming and certain cryptanalytic attacks, the sequences must have large linear spans.

There is a spectrum of families of sequences exhibiting tradeoffs among the size of the family, the maximum cross-correlations, and the linear span. For a family of size $2^n - 1$ with maximum correlation of $2^{n/2} - 1$, the size of the family can be at most $2^{n/2} - 1$. These values are achieved, for example, by bent function sequences [6, 7, 11], No sequences [9, 10], and TN sequences (also known as generalized No sequences) [4, 9].

If we want larger families of sequences, we must weaken the bound on the maximum correlations. A number of such families have been studied, such as Gold sequences and large sets of Kasami sequences [1, 2], but the linear spans of these sequences are only linear in n.

In this paper we describe new families of binary sequences which we call *QF sequences* (QF stands for "quadratic form"). The size of these families is $2^n + 1$ when the period is

*Project sponsored by the National Security Agency under Grant Number MDA904-91-H-0012. The United States Government is authorized to reproduce and distribute reprints notwithstanding any copyright notation hereon.

Family	n	Size of Family	Maximum Correlation	Maximum Linear Span	Range of Imbalance
Gold	$2m+1$	2^n+1	$1+2^{\frac{n+1}{2}}$	$2n$	$[1, 2^{\frac{n+1}{2}}+1]$
Gold	$4m+2$	2^n-1	$1+2^{\frac{n+2}{2}}$	$2n$	$[1, 2^{\frac{n+2}{2}}+1]$
Kasami (Small Set)	$2m$	$2^{\frac{n}{2}}$	$1+2^{\frac{n}{2}}$	$\frac{3n}{2}$	$[1, 2^{\frac{n}{2}}+1]$
Kasami (Large Set)	$4m+2$	$2^{\frac{n}{2}}(2^n+1)$	$1+2^{\frac{n+2}{2}}$	$\leq \frac{5n}{2}$	$[1, 2^{\frac{n+2}{2}}+1]$
Bent	$4m$	$2^{\frac{n}{2}}$	$1+2^{\frac{n}{2}}$	$\binom{n/2}{n/4}2^{\frac{n}{4}}$	1
No	$2m$	$2^{\frac{n}{2}}$	$1+2^{\frac{n}{2}}$	$m(2^m-1)$	$[1, 2^{\frac{n}{2}}+1]$
TN	$2km$	$2^{\frac{n}{2}}$	$1+2^{\frac{n}{2}}$	$> 3mk(3k-1)^{m-2}$	$[1, 2^{\frac{n}{2}}+1]$
$(0,j)$-QF	me	2^n+1	$1+2^{\frac{n+m}{2}}$	$\sim n2^{m-1}(e-2)^{m-2}$	$[1, 2^{\frac{n+m}{2}}+1]$

Table 1: COMPARISON OF PROPERTIES OF FAMILIES OF SEQUENCES OF PERIOD 2^n-1

2^n-1. Certain of these sequences, $(0,j)$-QF sequences, have maximum correlations that are slightly greater than $2^{n/2}-1$, and linear spans which are larger than Gold, Kasami, and GMW sequences by a factor that is exponential in n. These results are summarized in Table (1).

The construction of QF sequences is based on algebra over finite fields. They are special cases of the sequences known as d-form sequences [4]. The ith element of a QF sequence is given by starting with the ith power of a primitive element in the Galois field $GF(2^n)$, applying a quadratic form to $GF(2^m)$ (m a divisor of n), raising the result to some power, and mapping to $GF(2)$ by a trace function. We describe here the correlation and balance properties of $(0,j)$-QF sequences, based on previous results on TN sequences [4]. We also give lower bounds on the linear span of $(0,j)$-QF sequences, count the number of distinct families of such sequences, and discuss their implementation.

2 Definitions

Throughout this paper let e and m be positive integers and let $n = em$. For convenience we write $q = 2^m$. Let Tr_m^n be the trace function from $GF(2^n)$ to $GF(2^m)$, $Tr_m^n(x) = \sum_{i=0}^{e-1} x^{q^i}$. By a quadratic form on $GF(q)$ we mean a homogeneous polynomial of degree 2 in several variables.

By choosing coordinates, we can treat $GF(q^e)$ as e-dimensional affine space over $GF(q)$. Then any polynomial in e-variables with coefficients in $GF(q)$ defines a function from $GF(q^e)$ to $GF(q)$, and any such function is representable by a polynomial. Under this identification, the quadratic forms correspond precisely to sums of functions of the form

$$H(x) = Tr_m^n(\gamma x^t),$$

where the sum of the coefficients in the base q expansion of t is 2, $t < 1 + 2^{e/2}$, and $\gamma \in GF(q^e)$; and, if e is even, functions of the form

$$H(x) = Tr_m^{n/2}(\gamma x^{1+q^{e/2}}),$$

where $\gamma \in GF(q^{e/2})$.

QF sequences are generated in three steps. We start with a sequence of powers of a primitive element in $GF(q^e)$. To this sequence we apply a quadratic form H mapping to $GF(q)$. We then raise the result to some power. Finally, we apply the trace function Tr_2^m. The precise definition of QF sequences is as follows.

Definition 2.1 *Let e and m be positive integers, and let $q = 2^m$. Let r be a positive integer such that $\gcd(r, q-1) = 1$. Let α be a primitive element in $GF(q^e)$. Let $H(x)$ be a quadratic form on $GF(q^e)$ over $GF(q)$. Then the sequence $\mathbf{S}$ whose ith term is $s_i = Tr_1^e((H(\alpha^i))^r)$ is a* QF sequence.

QF sequences are a special case of d-form sequences, in which the quadratic form H is replaced by a homogeneous polynomial of degree d [4].

The particular QF sequences we are interested in here are those of the form

$$s_i = Tr_1^m((Tr_m^n(\gamma\alpha^{2i}) + Tr_m^n(\delta\alpha^{(1+q^j)i}))^r).$$

We call such a sequence a $(0, j)$-*QF sequence*. We require that $\gcd(e, j) = 1$. We sometimes also require that $\gcd(q^j + 1, q^e - 1) = 1$. If $\gcd(e, j) = 1$, this is equivalent to the condition that e is odd. We show that, for fixed n, m, j, and r, the family of all such $(0, j)$-QF sequences has near optimal cross-correlations and very high linear span. The number of cyclically distinct sequences in such a family is $2^n - 1$.

3 Cross-Correlations

In this section we determine the cross-correlations of the $(0, j)$-QF sequences defined in Section 2. Recall that the cross-correlation with shift τ of two sequences $\mathbf{S} = (s_1, s_2, \cdots)$ and $\mathbf{T} = (t_1, t_2, \cdots)$ of period N, is defined by

$$\Theta_{\mathbf{S},\mathbf{T}}(\tau) = \sum_{i=1}^{N} (-1)^{s_i + t_{i+\tau}}. \tag{1}$$

In our case $N = q^e - 1$. The cross-correlations of two d-form sequences have been described in terms of the zeros of d-forms [4].

Theorem 3.1 (Klapper [4]) *Let the integers m, e, and r, and primitive element $\alpha \in GF(q^e)$ be fixed, and let H_1 and H_2 be d-forms on $GF(q^e)$ over $GF(q)$ defining d-form sequences $\mathbf{S}^1$ and $\mathbf{S}^2$. For any shift τ, let $z_\tau = |\{x \neq 0 \in GF(q^e) : H_1(x) + H_2(\alpha^\tau x) = 0\}|$. Then*

$$\Theta_{\mathbf{S}^1,\mathbf{S}^2}(\tau) = \frac{qz_\tau - (q^e - 1)}{q - 1}.$$

In case $d = 2$ and H is a quadratic form, the number of zeros of H is well understood. Recall that the *rank* of a quadratic form H is the smallest integer t such that there is a set of coordinates in which H can be represented using only t variables. The number of solutions x to the equation $H(x) = 0$ (or, more generally, $H(x) = a$) is determined by the rank and, in the case of even rank, whether H is of one of two types. A nice treatment of this analysis can be found in Lidl and Neiderreiter's book [8].

Let r, α, and j be fixed satisfying $\gcd(r, q-1) = 1$. Consider a pair of $(0, j)$-QF sequences

$$s_i^k = Tr_1^m((Tr_m^n(\gamma_k\alpha^{2i}) + Tr_m^n(\delta_k\alpha^{i(1+q^j)}))^r),$$

where $\gamma_k, \delta_k \in GF(q^n)$, and $k = 1, 2$. To compute the cross-correlations of $\mathbf{S}^1$ and $\mathbf{S}^2$, we need only count the number z_τ of zeros of the quadratic form

$$Tr_m^n((\gamma_1 + \gamma_2\alpha^{2\tau})x^2) + Tr_m^n((\delta_1 + \delta_2\alpha^{(1+q^j)\tau})x^{1+q^j}) \stackrel{\text{def}}{=} Tr_m^n(\gamma x^2) + Tr_m^n(\delta x^{1+q^j}),$$

which we denote by $H(x)$.

The number of solutions to $H(x) = 0$ is the sum over $v \in GF(q)$ of the number of solutions $N(v)$ to the system

$$Tr_m^n(\beta x) = v \text{ and } Tr_m^n(\delta x^{1+q^j}) = v^2, \tag{2}$$

where $\beta^2 = \gamma$. $N(v)$ can be counted using the techniques of [5]. The function $\eta(v)$ is defined by $\eta(v) = -1$ if $v \neq 0$, and $\eta(0) = q - 1$. We have

Proposition 3.2 *Let* $\gcd(e, j) = 1$. *If* e *is even, then for every* $\gamma \neq 0$ *and* $\delta \neq 0$ *there is an* $\epsilon \in \{\pm 1\}$ *and* $t \in \{e/2 - 2, e/2 - 1\}$ *such that one of the following holds for every* $v \in GF(q)$:

1. $N(v) = q^{e-2} + \epsilon\eta(v)q^t$.
2. $N(0) = q^{e-2}$ *and* $N(v) = q^{e-2} + \epsilon q^{t+1}$ *if* $v \neq 0$.
3. $N(0) = q^{e-2} + \epsilon(q-1)q^{t+1}$ *and* $N(v) = q^{e-2}$ *if* $v \neq 0$.

If e *is odd, then for every* $\gamma \neq 0$ *and* $\delta \neq 0$ *there is an* $\epsilon \in \{\pm 1\}$ *such that one of the following holds for every* $v \in GF(q)$:

1. $N(v) = q^{e-2}$.
2. $N(v) = q^{e-2} + \epsilon\eta(v)q^{(e-3)/2}$.
3. $N(v) = q^{e-2} + \epsilon(q-1)q^{(e-3)/2}$.

These results can be combined with Theorem 3.1 to give the cross-correlations of $(0, j)$-QF sequences. If $\mathbf{S}$ is a cyclic shift of $\mathbf{T}$, and τ is a shift so that $\mathbf{S}$ coincides with $\mathbf{T}$ shifted by τ, then the cross-correlation of $\mathbf{S}$ and $\mathbf{T}$ with shift τ is $2^n - 1$. We call this a *trivial* cross-correlation.

Theorem 3.3 *Let* j *and* r *be integers satisfying* $\gcd(e, j) = \gcd(r, q-1) = 1$. *Let* $\mathbf{S}$ *and* $\mathbf{T}$ *be two* $(0, j)$*-QF sequences with exponent* r.

1. *If* e *is even, then the nontrivial cross-correlations of* $\mathbf{S}$ *and* $\mathbf{T}$ *are* $\{-1, \pm q^{e/2} - 1, \pm q^{e/2+1} - 1\}$.
2. *If* e *is odd then the nontrivial cross-correlations of* $\mathbf{S}$ *and* $\mathbf{T}$ *are* $\{-1, \pm q^{(e+1)/2} - 1\}$.

The imbalance of a binary sequence is the number of times the sequence equals zero minus the number of times it equals one. The imbalances of $(0, j)$-QF sequences are in the same set of values as the cross-correlations.

Corollary 3.4 *Let* j *and* r *be integers satisfying* $\gcd(e, j) = \gcd(r, q-1) = 1$ *and* $\mathbf{S}$ *be a* $(0, j)$*-QF with exponent* r.

1. *If* e *is even, then the imbalance of* $\mathbf{S}$ *is in the set* $\{-1, \pm q^{e/2} - 1, \pm q^{e/2+1} - 1\}$.
2. *If* e *is odd then the the imbalance of* $\mathbf{S}$ *is in the set* $\{-1, \pm q^{(e+1)/2} - 1\}$.

We can also use Theorem 3.1 to show that the period of a quadratic form sequence is $2^n - 1$.

4 Linear Span

In this section we show that the linear span, $\lambda_{\mathbf{S}}$, of a $(0,j)$-QF sequence $\mathbf{S}$ is at least that of a GMW sequence with the same period, and that when γ and δ are nonzero, $\lambda_{\mathbf{S}}$ asymptotically exceeds the linear span of a GMW sequence by a factor of $2^{\mathrm{wt}(r)}$, where $\mathrm{wt}(r)$ is the number of ones in the binary expansion of r. If $\gamma = 0$ or $\delta = 0$, then $\mathbf{S}$ is a GMW sequence and, as is well known, the linear span is $e^{\mathrm{wt}(r)}$.

The binary expansion of the exponent r in the definition of a $(0,j)$-QF sequence is a series of runs of ones separated by zeros. We let U be the number of runs of ones and let R_i be the length of the ith run.

Theorem 4.1 *Let* $\mathbf{S}$ *be a* $(0,j)$*-QF sequence with*

$$s_i = Tr_1^m((Tr_m^n(\gamma\alpha^i)^2 + Tr_m^n(\delta\alpha^{i(1+q^j)}))^r).$$

Suppose γ *and* δ *are nonzero. Let* $\pi = (e^2 - 4e + 1)^{1/2}$*. If* $e > 3$*, then the linear span of* $\mathbf{S}$ *is at least*

$$m\prod_{i=1}^{U}\left(\frac{(e+1+\pi)(e-1+\pi)^{R_i} - (e-1-\pi)(e+1-\pi)^{R_i}}{2\pi}\right)$$

$$\geq \begin{cases} 2n\cdot 4^{\mathrm{wt}(r)-1} & \text{if } e = 4 \\ n\cdot\prod_{i=1}^{U}(2^{R_i}(e-2)^{R_i-1} - 2\cdot 4^{R_i-1}) & \text{otherwise.} \end{cases}$$

This last product is maximized when $r = 2^{m-1} - 1$ *at* $n(2^{m-1}(e-2)^{m-2} - 2\cdot 4^{m-2})$*. If* $e = 3$*, then the linear span of* $\mathbf{S}$ *is at least* $m6^U 4^{\mathrm{wt}(r)}$*. This is maximized at* $m\cdot 6^{m/2}\cdot 4^{m/2}$ *when* m *is even by taking* $r = 1+2^2+2^4+\cdots 2^{m-2}$*. It is maximized at* $m\cdot 6^{(m-1)/2}\cdot 4^{(m+1)/2}$ *when* m *is odd by taking* $r = 1+2+2^3+2^5+\cdots 2^{m-2}$*.*

Proof: Key [3] showed that if we express the ith term of a sequence $\mathbf{S}$ as a polynomial in α^i, then $\lambda_{\mathbf{S}}$ is the number of monomials in that polynomial. Equivalently, the linear span is the number of monomials in the polynomial

$$s(x) \;=\; Tr_1^m((Tr_m^n(\gamma x)^2 + Tr_m^n(\delta x^{1+q^j}))^r).$$

We show that, when this expression is expanded to a sum of monomials, all but a small number of monomials are distinct. We explicitly describe those monomials that coincide with others, and count the remaining monomials.

Let the binary expansion of r be $r = \sum_{\ell=0}^{m-1} a_\ell 2^\ell$, $a_\ell \in \{0,1\}$. Let $L = \{\ell : a_\ell \neq 0\}$. Let Δ be the set of $\mathrm{wt}(r)$-tuples of elements of $\{0,\cdots,e-1\}$, indexed by L, and let Γ be the the set of $\mathrm{wt}(r)$-tuples of elements of $\{2, 1+q^j\}$, indexed by L. For $I \in \Delta$ and $K \in \Gamma$, we write i_ℓ and k_ℓ for the ℓth components of I and K, respectively.

Lemma 4.2 *The polynomial* $s(x)$ *can be expanded to a sum of monomials of the form* $x^{\mu(t,I,K)}$*, where*

$$\mu(t,I,K) = 2^t\sum_{\ell\in L} 2^\ell k_\ell q^{i_\ell},$$

$0 \leq t < m$*,* I *ranges over* Δ*, and* K *ranges over* Γ*.*

We use the following terminology. Each $\mu(t, I, K)$ is an *exponent*. It is a sum of *terms* $2^\ell k_\ell q^{i_\ell}$, with $k_\ell \in \{2, 1+q^j\}$. Each term is a sum of one or two *summands*, $2^{\ell+1}q^{i_\ell}$; or $2^\ell q^{i_\ell}$ and $2^\ell q^{i_\ell+j}$.

Let $\mu(t, I, K)$ and $\mu(t', I', K')$ be two exponents. We say (t, I, K) is *equivalent* to (t', I', K'), written $(t, I, K) \sim (t', I', K')$, if

$$\mu(t, I, K) \equiv \mu(t', I', K') \bmod q^e - 1. \tag{3}$$

Two index triples are equivalent if and only if they correspond to the same powers of x, but with possibly different coefficients. Their resulting monomials may (or may not) cancel. Thus we want to count the number of (t, I, K) that are equivalent to other index triples, and subtract this number from the total number of index triples. This gives us a lower bound on the number of monomials in $s(x)$ with nonzero coefficients, and so a lower bound on $\lambda_{\mathbf{S}}$. The only exponents that are equivalent to other exponents are those containing consecutive terms of one of the following six types:

1. $k_\ell = 2$, $k_{\ell+1} = 1 + q^j$, $i_{\ell+1} = i_\ell$;
2. $k_\ell = 2$, $k_{\ell+1} = 1 + q^j$, $i_{\ell+1} = i_\ell - j$;
3. $k_\ell = 2$, $k_{\ell+1} = 2$, $i_{\ell+1} = i_\ell - j$;
4. $k_\ell = 2$, $k_{\ell+1} = 2$, $i_{\ell+1} = i_\ell + j$;
5. $k_\ell = 2$, $k_{\ell+1} = 1 + q^j$, $i_{\ell+1} = i_\ell + j$;
6. $k_\ell = 2$, $k_{\ell+1} = 1 + q^j$, $i_{\ell+1} = i_\ell - 2j$.

Thus the linear span of $\mathbf{S}$ is at least the number of exponents with no such consecutive terms of these types. We call these *good* exponents. Note that if $e = 3$, then $i_\ell - 2j \equiv i_\ell + j \bmod e$, so the fifth and sixth types coincide and there is no distinct pair of equivalent exponents. Otherwise all six types are distinct.

Any exponent for which all $k_\ell = 1 + q^j$ is good. Thus there are at least $e^{\mathrm{wt}(r)}$ good exponents for each t, $0 \le t < m$, and the linear span is at least $me^{\mathrm{wt}(r)}$, the linear span of a GMW sequence of period $2^n - 1$ with intermediate field $GF(q)$ and exponent r.

Consider a single run of ones of length R in the binary expansion of r. We can count the number of (i_ℓ, k_ℓ) for ℓ in this run that give good exponents by building them up a term at a time, from small ℓ to large ℓ. Given an exponent that is good up to its ℓth term, we count the ways it can be extended to a good exponent by adding a term for the next bit of r. If the next bit of r is the first in a run of ones, or if $k_\ell = 1 + q^j$, then there are no restrictions, so there are $2e$ ways of extending. Otherwise, there are $e - 4$ ways of extending the exponent with $k_{\ell+1} = 1 + q^j$ (one way if $e = 3$), and $e - 2$ ways of extending it with $k_{\ell+1} = 2$.

It follows that the good choices of $\{k_\ell\}$ and $\{i_\ell\}$ for one run are independent of those for other runs. The total number of good exponents is therefore the product of the number of good choices for the different runs. For each run, we can count the good choices by a recurrence. Only the length of the run matters, so we can assume the run is a sequence of consecutive indices $0, 1, 2, \cdots, R-1$. We let $A(\ell)$ be the number of good runs up to index ℓ, ending with $k_\ell = 2$, and let $B(\ell)$ be the number of good runs up to index ℓ, ending with $k_\ell = 1 + q^j$. Then, for $e > 3$, we have the system of recurrences

$$\begin{aligned} A(\ell) &= (e-2)A(\ell-1) + eB(\ell-1) \\ B(\ell) &= (e-4)A(\ell-1) + eB(\ell-1), \end{aligned} \tag{4}$$

for $1 \le \ell < R$. For $e = 3$, we have the system

$$\begin{aligned} A(\ell) &= A(\ell-1) + 3B(\ell-1) \\ B(\ell) &= A(\ell-1) + 3B(\ell-1), \end{aligned}$$

for $1 \le \ell < R$. In both cases the initial conditions are $A(0) = B(0) = e$.

First consider the case when $e = 3$. We have $A(\ell) = B(\ell)$ for all ℓ, so $A(\ell) = 4A(\ell-1)$. The solution to this recurrence is $A(\ell) = 3 \cdot 4^\ell$, so $A(\ell) + B(\ell) = 6 \cdot 4^\ell$. Thus the total number of good exponents for a run of length R when $e = 3$ is $6 \cdot 4^R$. The total number of good exponents for a given r is the product

$$\prod_{i=1}^{U} 6 \cdot 4^{R_i} = 6^U \cdot 4^{\mathrm{wt}(r)},$$

where i ranges over the set of runs of ones in the binary expansion of r, R_i is the length of the ith run, and U is the number of runs. For each good exponent $\mu(I, K)$ and each t, we have a monomial that must appear in $s(x)$ with a nonzero coefficient. Thus the linear span is at least $m6^U \cdot 4^{\mathrm{wt}(r)}$. This is maximized at $m \cdot 6^{m/2} \cdot 4^{m/2}$ when m is even by taking $r = 1 + 2^2 + 2^4 + \cdots 2^{m-2}$. It is maximized at $m \cdot 6^{(m-1)/2} \cdot 4^{(m+1)/2}$ when m is odd by taking $r = 1 + 2 + 2^3 + 2^5 + \cdots 2^{m-2}$.

The general case is somewhat more complicated. Equation (4) can be interpretted as a matrix equation,

$$\begin{pmatrix} A(\ell) \\ B(\ell) \end{pmatrix} = \begin{pmatrix} e-2 & e \\ e-4 & e \end{pmatrix} \cdot \begin{pmatrix} A(\ell-1) \\ B(\ell-1) \end{pmatrix} \stackrel{\text{def}}{=} M \cdot \begin{pmatrix} A(\ell-1) \\ B(\ell-1) \end{pmatrix}. \tag{5}$$

Equation (5) is solved by diagonalizing M. Let $\pi = (e^2 - 4e + 1)^{1/2}$. Then $M = D^{-1}ND$, where

$$D = \begin{pmatrix} -1+\pi & e \\ -1-\pi & e \end{pmatrix},$$

and

$$N = \begin{pmatrix} e+\pi-1 & 0 \\ 0 & e-\pi+1 \end{pmatrix}.$$

The total number of good exponents for a run of length R is therefore given by

$$\begin{aligned} \begin{pmatrix} 1 & 1 \end{pmatrix} M^{R-1} \begin{pmatrix} e \\ e \end{pmatrix} &= \begin{pmatrix} 1 & 1 \end{pmatrix} D^{-1} N^{R-1} D \begin{pmatrix} e \\ e \end{pmatrix} \\ &= \left(\frac{e+1+\pi}{2\pi} (e-1+\pi)^R - \frac{e-1-\pi}{2\pi} (e+1-\pi)^R \right) \end{aligned}$$

The total number of good exponents for a given r is therefore the product

$$\prod_i \left(\frac{e+1+\pi}{2\pi} (e-1+\pi)^{R_i} - \frac{e-1-\pi}{2\pi} (e+1-\pi)^{R_i} \right).$$

This is maximized when $\mathrm{wt}(r) = m-1$, and r has a single run of ones. In this case the linear span is at least

$$\left(\frac{e+1+\pi}{2\pi}(e-1+\pi)^{m-1} - \frac{e-1-\pi}{2\pi}(e+1-\pi)^{m-1}\right).$$

The estimates in the statement of the theorem follow from the observations that $e-3 < \pi < e-2$ when $e \geq 5$, and that $\pi = 1$ when $e = 4$. □

5 The Number of Distinct Families of $(0, J)$-QF Sequences

It is useful to know how many distinct families of $(0, j)$-QF sequences there are with a given period $2^n - 1$. In this section we keep n fixed and let the factorization $n = me$ vary. We show that each choice of parameters m (dividing n, with $e = n/m$), j (relatively prime to e and less than $e/2$), r (up to multiplication by a power of 2), and α (up to raising to an exponent which is a power of two) gives rise to a distinct family of sequences, in the sense that no sequence in one family is a cyclic shift of a sequence in another family. For any fixed even integer n we write

$$S(m, j, \alpha, r) = \{\mathbf{S}^{\gamma,\delta} : s_i = Tr_1^m((Tr_m^n(\gamma\alpha^{2i} + \delta\alpha^{i(1+2^{mj})}))^r) \text{ and } \gamma, \delta \in GF(2^k)\},$$

where m divides n, $e = n/m$, $\gcd(j, e) = 1$, $j < e/2$, α is a primitive element of $GF(2^n)$, and r is relatively prime to $2^m - 1$. The family does not change when r is multiplied by a power of two, so we can assume that $r < 2^{m-1}$ and is odd. Note that a cyclic shift of one of these sequences by τ places replaces γ by $\gamma\alpha^{2\tau}$, and replaces δ by $\delta\alpha^{\tau(1+2^{mj})}$, and hence gives another sequence in the family.

Proposition 5.1 *Let n be a positive integer, $N = 2^n - 1$, m_1, and m_2 be divisors of n such that m_1 and m_2 divide n, and $e_1 = n/m_1$ and $e_1 = n/m_1$. Let r_1, r_2, j_1, and j_2 be integers such that $1 \leq r_k < 2^{m_k - 1}$, r_k is relatively prime to $2^{m_k} - 1$, $j_k < e_k/2$, and j_k is relatively prime to e_k. Let α_1 and α_2 be primitive elements in $GF(2^n)$. Then $S(m_1, j_1, \alpha_1, r_1)$ and $S(m_2, j_2, \alpha_2, r_2)$ are disjoint unless either*

1. *$m_1 = m_2$, $j_1 = j_2$, and for some integers u and v, $0 \leq u < n$, and $0 \leq v < m_1$, $\alpha_2 = \alpha_1^{2^u}$, and $r_1 = 2^v \cdot r_2$, or*
2. *$m_1 j_1 = m_2 j_2$, r_1 and r_2 are powers of 2, and for some integer u, $0 \leq u < n$ and $\alpha_2 = \alpha_1^{2^u}$.*

In each of these cases

$$S(m_1, \alpha_1, r_1) = S(m_2, \alpha_2, r_2).$$

It is now straightforward to determine the number of distinct families of $(0, j)$-QF sequences.

Theorem 5.2 *Let n be a nonnegative integer. The number $\mathbf{N}_{\mathrm{QF}}$ of distinct families of $(0, j)$-QF sequences of period $2^n - 1$ is given by*

$$\mathbf{N}_{\mathrm{QF}} = \frac{\phi(2^n - 1)}{n} \cdot \left(\sum_{e|n, e>1} \frac{\phi(e)}{2}\left(\frac{e\phi(2^{n/e} - 1)}{n} - 1\right) + \frac{\phi(n)}{2}\right),$$

where $\phi(\cdot)$ is Euler's phi function.

If n is not a power of 2, The number $\mathbf{N}'_{\mathrm{QF}}$ of distinct families of $(0, j)$-QF sequences of period $2^n - 1$ with correlations in $\{\pm q^{(n+1)/2} - 1, -1\}$ is given by

$$\mathbf{N}'_{\mathrm{QF}} = \frac{\phi(2^n - 1)}{n} \cdot \left(\sum_{\substack{e \mid n \\ e > 1 \ odd}} \frac{\phi(e)}{2} \left(\frac{e\phi(2^{n/e} - 1)}{n} - 1 \right) + \frac{\phi(n')}{2} \right),$$

where n' is the maximal odd divisor of n.

References

[1] T. Kasami, "Weight distribution formula for some classes of cyclic codes," Coordinated Science Laboratory, University of Illinois, Urbana, Tech. Rep. R-285 (AD632574), 1966.

[2] T. Kasami, "Weight distribution of Bose-Chaudhuri-Hocquenghem codes," in *Combinatorial Mathematics and its Applications.* Chapel Hill, NC: University of North Carolina Press, 1969.

[3] E. L. Key, "An Analysis of the structure and complexity of nonlinear binary sequence generators," *IEEE Trans. Info. Theory,* vol. IT-22 no. 6, pp. 732-736, Nov. 1976.

[4] A. Klapper, "d-form Sequences: Families of Sequences with Low Correlation Values and Large Linear Span," University of Kentucky, Department of Computer Science Technical Report #240-93, 1993.

[5] A. Klapper, "Cross-correlations of geometric sequences in characteristic two," *Designs, Codes, and Cryptography*, vol. 3, pp. 347-377, 1993.

[6] P. V. Kumar and R. A. Scholtz, "Bounds on the linear span of bent sequences," *IEEE Trans. Info. Theory,* vol. IT-29, pp. 854-862, Nov. 1983.

[7] A. Lempl and M. Cohn, "Maximal families of bent sequences," *IEEE Trans. Info. Theory,* vol. IT-28, pp. 865-868, Nov. 1982.

[8] R. Lidl and H. Niederreiter *Finite Fields* in *Encyclopedia of Mathematics Vol. 20,* Cambridge University Press: Cambridge, 1983.

[9] J. No, *A new family of binary pseudorandom sequences having optimal periodic correlation properties and large linear span,* Doctoral Dissertation, University of Southern California, 1988.

[10] J. No and P. V. Kumar, "A new family of binary pseudorandom sequences having optimal periodic correlation properties and large linear span," *IEEE Trans. on Info. Theory*, vol. 35, pp. 371-379, 1989.

[11] O. Rothaus, "On bent functions," *Journal of Combinatorial Theory Series A*, vol. 20, pp. 300-305, 1976.

[12] M. Simon, J. Omura, R. Scholtz, and B. Levitt, *Spread-Spectrum Communications Vol. 1*, Computer Science Press: 1985.

GENERATING FUNCTION BOUNDS FOR CONCATENATED TRELLIS AND REED-SOLOMON CODING [1]

John S. Sadowsky
Arizona State University
Tempe, AZ 85287-7206

ABSTRACT

This paper develops a new generating function bound for the outer code decoding error probability of a concatenated coding system consisting of a trellis inner code, a Reed-Solomon outer code, and a symbol interleaver between the inner and outer codes. The generating functions derived here enumerate only those inner code error events of sufficient length to result in an outer Reed-Solomon decoding error. We fully account for the possibility of an outer decoding error due to multiple inner code error events, and we fully account for dependency in successive Reed-Solomon code symbol errors; that is, we make no ideal interleaving assumption.

1. INTRODUCTION

In this paper we derive new generating functions for bounding the outer decoding error probability for a concatenated coding system. Our system consists of a trellis inner code, a symbol interleaver, and a Reed-Solomon outer code. This is a standard configuration, especially in space and satellite communications.

The standard theory must be modified in a number of ways. First, rather than counting all of the trellis code's error events, we derive new generating functions that enumerate only those error events that cause error bursts of sufficient length to result in an outer Reed-Solomon decoding error. A given error burst of sufficient length, however, may or may not cause a outer decoding error depending on its *position* within the scope of the target Reed-Solomon codeword. Thus, the information weights must account for both error burst length and position. In addition, an outer Reed-Solomon decoding error may result from the combination of several Viterbi decoder error events. We do not assume ideal interleaving (i.e., independent Reed-Solomon code symbol errors).

The basic theory of these bounds was developed by Viterbi. See [4]. Zehavi and Wolf [5] derived an extension to a certain class of trellis codes based on rate $(n-1)/n$ convolutional encoders, but the techniques presented here apply to any "quasi-regular" as defined in [1]. These codes includes the Ungerboeck codes [2], and the pragmatic trellis codes of [3].

2. PRELIMINARIES

We will hereafter use TC and RS to abbreviate, respectively, "trellis code" and "Reed-Solomon." A list of some system parameters includes b_{RS} = the number of bits per RS symbol, n_{RS} = the RS codeword length in RS code symbols, t_{RS} = the number of RS code symbol errors correctable by the RS code, n_{TC} = the number of TC states, b_{TC} =

[1]This work has been supported by the Fairchild Data Corporation, 350 North Hayden Road, Scottsdale, AZ 85257.

the number of information bits in a TC information symbol, $\widetilde{b}_{TC}$ = the number of code bits in a TC code symbol, $r_s = b_{RS}/b_{TC}$ (the symbol ratio), and d_I = the interleaver depth. The terms *code symbol* and *information symbol* are applied to both the inner and outer codes. We assume that b_{RS} is an integral multiple of b_{TC}, so r_s is an integer, and that TC information symbols are packed into RS code symbols in such a way that TC information symbols do not straddle two RS code symbols. Thus, one RS code symbol contains exactly r_s TC information symbols.

Our goal is to evaluate $P_e \stackrel{\text{def}}{=}$ the probability that a target Reed-Solomon codeword is incorrectly decoded. Inner Viterbi decoder error events are the root cause of error. We distinguish between an *error event*, which is a divergence of the correct and decoded paths on the trellis diagram, and the resulting *TC error bursts*, which is the corrupted sequences of TC information symbols at the output of the Viterbi decoder. After de-interleaving, a TC error burst in turn results in an *RS error burst*, which is a sequence of corrupted RS code symbols at the input of the outer RS decoder. Thus, our analysis is based on the causality relationship

Viterbi decoder error events	$\Rightarrow$	TC error bursts	$\Rightarrow$	RS error bursts	$\Rightarrow$	target RS codeword decoding error

Quasi-regular trellis codes consist of a linear (binary) convolutional code followed by a signal map. Encoder states are labeled $s = 0, 1, .., n_{TC} - 1$. This labeling of states is arbitrary, except that $s = 0$ is reserved for the zero state. On the trellis diagram, a Viterbi decoder *error event* is determined by a *correct path* $(s_0, \mathbf{x})$ and an *error path* $(s_0, \mathbf{x}+\mathbf{e})$. Both $\mathbf{e} = (e_1, .., e_L)$ and $\mathbf{x} = (x_1, .., x_L)$ are sequences of TC information symbols that determine the trellis paths emanating from the common *initial state* s_0. We call $\mathbf{e}$ the *error sequence*, and $L = L(\mathbf{e})$ is its length in trellis branches. For a legitimate error event, both paths diverge from s_0, merge into a common final state after the Lth branch, and do not simultaneously encounter the same state in between. By the linearity of the convolutional code, $(s_0, \mathbf{x}+\mathbf{e})$ is a legitimate error event error path relative to any correct path $(s_0, \mathbf{x})$ if and only if $(0, \mathbf{e})$ is a legitimate error path relative to the all zero correct path $(0,\mathbf{0})$. The correct path $(s_0, \mathbf{x})$ is arbitrary, hence, there are $n_{TC}\, 2^{b_{TC}L(\mathbf{e})}$ possible correct paths for a given error sequence $\mathbf{e}$. Notice that the above error event specification does not include *position*, that is, the stage of the trellis where the decoded path and correct paths diverge from state s_0.

Let $\widetilde{\mathbf{x}}$ and $\widetilde{\mathbf{e}}$ denote the sequences of TC code symbol sequences, respectively, for paths $(s_0, \mathbf{x})$ and $(0, \mathbf{e})$. Then, by linearity, $\widetilde{\mathbf{x}} + \widetilde{\mathbf{e}}$ is the TC code symbol sequence for the error path $(s_0, \mathbf{x} + \mathbf{e})$. Let $y(\cdot)$ denote the signal map. Then $d_E(s_0, \mathbf{x}, \mathbf{e}) \stackrel{\text{def}}{=} \sum_{k=1}^{L(\mathbf{e})} \|y(\widetilde{x}_k + \widetilde{e}_k) - y(\widetilde{x}_k)\|^2$ is the *squared Euclidean distance* between the correct and error paths.

For an error event error sequence $\mathbf{e}$, define $P(\mathbf{e})$ to be the probability that error sequence $\mathbf{e}$ is decoded starting in any fixed position. The *error sequence information weight* $a(d;\mathbf{e})$ is defined to be the fraction of the $n_{TC}2^{b_{TC}L(\mathbf{e})}$ possible correct paths $(s_0, \mathbf{x})$ such that $d_E(s_0, \mathbf{x}, \mathbf{e}) = d$. The fundamental union bound is

$$P(\mathbf{e}) \;=\; \frac{1}{n_{TC}2^{b_{TC}L(\mathbf{e})}} \sum_{(s_0,\mathbf{x})} \mathcal{P}(\, E(s_0, \mathbf{x}, \mathbf{e})\,) \;\leq\; \sum_{d} a(d;\mathbf{e})\, Q\left(\sqrt{\frac{d\,\mathcal{E}_s}{2N_0}}\right). \quad (1)$$

Quasi-regular codes have the property that the error sequence information weight $a(d;\mathbf{e})$ is a sum of branch-by-branch products of single branch information weights [1, 5].

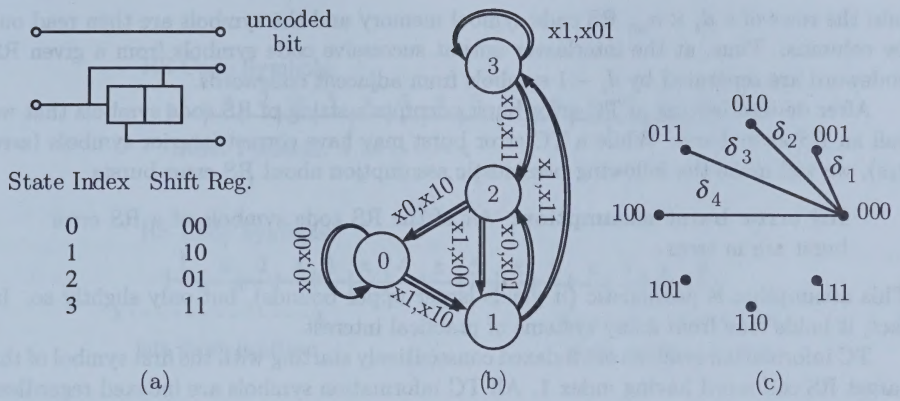


Figure 1: (a) A 4 state 8PSK trellis code, (b) the state diagram, and (c) the 8PSK signal map. The symbol x indicates the uncoded bit.

Put $\mathcal{D}(\tilde{e}) \stackrel{\text{def}}{=} \{\|y(\tilde{x}+\tilde{e})-y(\tilde{x})\|^2\}$ where $\tilde{x}$ ranges over all possible $2^{\tilde{b}_{TC}}$ TC code symbols. For $d \in \mathcal{D}(\tilde{e})$, let $\alpha(d;\tilde{e})$ be the fraction of the $2^{\tilde{b}_{TC}}$ code symbols $\tilde{x}$ such that $\|y(\tilde{x}+\tilde{e})-y(\tilde{x})\|^2 = d$. Let $\mathbf{e}$ be any fixed error sequence. ($\mathbf{e}$ does not need to be a full error event error sequence, that is, the path (0,$\mathbf{e}$) need not be merged back into the 0 state.) For each single branch error $\tilde{e}$ define the *weight profile* $\psi(D;\tilde{e}) \stackrel{\text{def}}{=} \sum_{d\in\mathcal{D}(\tilde{e})} \alpha(d;\tilde{e})\, D^d$. Then, for a quasi-regular code, the path distance transform is

$$\Psi(D;\mathbf{e}) \stackrel{\text{def}}{=} \sum_{d} a(d;\mathbf{e})\, D^d = \prod_{k=1}^{L(\mathbf{e})} \psi(D;\tilde{e}_k). \tag{2}$$

In the standard theory, the code's generating function is obtained as the transfer function of an "error event enumeration diagram." This diagram is simply the code's state diagram with the zero state split into an initial and a final absorbing state and branches labeled with branch weight profiles. The paths through the error event enumeration diagram are the error paths $(0,\mathbf{e})$. Correct path variation is accounted for by weight profiles.

In later sections, we will use the 8PSK trellis code of Figure 1 as an example. This code has an uncoded information bit that results in parallel branches for each state transition $\sigma \longrightarrow s$. Figure 1(c) gives the 8PSK signal map and also illustrates the possible squared Euclidean distances which we denote $\delta_1, .., \delta_4$.

Next, for an error event error sequence $\mathbf{e}$ define

$$\ell(\mathbf{e}) \stackrel{\text{def}}{=} \max\{k:\ e_k \neq 0\}. \tag{3}$$

Then the *TC error burst* caused by an error event with error sequence $\mathbf{e}$ is just $(e_1, .., e_{\ell(\mathbf{e})})$. Notice that the first and last symbols of a TC error burst are true errors: $e_1 \neq 0$ and $e_{\ell(\mathbf{e})} \neq 0$. However, we may have $e_k = 0$ for some $1 < k < \ell(\mathbf{e})$.

The interleaver lies between the inner and outer encoders, and it operates on RS code symbols. It can be either a block or convolutional interleaver. RS codewords are written

into the rows of a $d_I \times n_{RS}$ RS code symbol memory and the symbols are then read out by columns. Thus, at the interleaver output successive code symbols from a given RS codeword are separated by $d_I - 1$ symbols from adjacent codewords.

After de-interleaving, a TC error burst corrupts a string of RS code symbols that we call an *RS error burst*. While a TC error burst may have correct interior symbols (zero e_ks), we will make the following pessimistic assumption about RS error bursts.

> **RS error burst assumption:** *All* of the RS code symbols of a RS error burst are in error.

This assumption is pessimistic (it yields larger upper bounds), but only slightly so. In fact, it holds true from many systems of practical interest.

TC information symbols are indexed consecutively starting with the first symbol of the target RS codeword having index 1. All TC information symbols are indexed regardless of whether they are from the target codeword or adjacent interleaved codewords. On the other hand, we only index the RS code symbols of the target code words, and these have indices $p = 1, .., n_{RS}$. Thus, the TC information symbols in the pth RS code symbol have indices $i = (p-1)d_I r_s + 1, .., [(p-1)d_I + 1]r_s$. The *position* of a TC or RS error burst is the position of its leading symbol error.

3. THE SINGLE ERROR EVENT CASE

A single TC error event will result in an outer decoding error only if it causes an RS error burst of length $> t_{RS}$. Define $n_e(\ell; n_{RS}, t_{RS})$ to be the number of positions of a length ℓ TC error burst that will cause an RS error burst of more than t_{RS} RS symbols. A key parameter is

$$\ell^*(t) \stackrel{\text{def}}{=} \max\{\ell : n_e(\ell; n, t) = 0\} = (d_I t - 1) r_s + 1. \tag{4}$$

The single error event union bound is

$$P_{e1}(n_{RS}, t_{RS}) \leq \sum_{\mathbf{e}} n_e(\ell(\mathbf{e}); n_{RS}, t_{RS})\, P(\mathbf{e})$$

where $\mathbf{e}$ ranges over all error event error sequences (but $n_e(\ell(\mathbf{e}); n_{RS}, t_{RS}) = 0$ for some of the terms above). Applying (1) yields

$$P_{e1}(n_{RS}, t_{RS}) \leq \sum_{\ell} \sum_{d} n_e(\ell; n_{RS}, t_{RS})\, c(\ell, d)\, Q\left(\sqrt{\frac{d\,\mathcal{E}_s}{2N_0}}\right)$$

where $c(\ell, d) \stackrel{\text{def}}{=} \sum_{\mathbf{e}:\, \ell(\mathbf{e}) = \ell} a(d; \mathbf{e})$. Next, define $D^* \stackrel{\text{def}}{=} e^{-(\mathcal{E}_S/4N_0)}$, $d_f(t) \stackrel{\text{def}}{=} \min_{s_0, \mathbf{x}, \mathbf{e}:\, \ell(\mathbf{e}) > \ell^*(t)} d_E(s_0, \mathbf{x}, \mathbf{e})$, and $C(t) \stackrel{\text{def}}{=} \sqrt{N_0/(\pi\, \mathcal{E}_s\, d_f(t))}$. Then for every term in the last $P_{e1}(n_{RS}, t_{RS})$ bound we have $Q(\sqrt{d\,\mathcal{E}_s/N_0}) \leq C(t_{RS})\,(D^*)^d$, and hence, the modified union bound is

$$P_{e1}(n_{RS}, t_{RS}) \leq C(t_{RS}) \sum_{\ell = \ell^*(t_{RS})+1}^{\infty} \sum_{d} n_e(\ell; n_{RS}, t_{RS})\, c(\ell, d)\, (D^*)^d. \tag{5}$$

Next we eliminate the factor $n_e(\ell; n_{RS}, t_{RS})$. Observe from definition (4) that TC error burst of length ℓ in the range $\ell^*(t) < \ell \leq \ell^*(t+1)$ will corrupt either t or $t+1$ RS

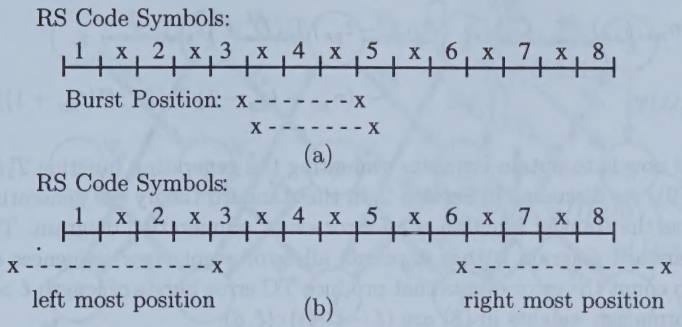

RS Code Symbols:
1 x 2 x 3 x 4 x 5 x 6 x 7 x 8
Burst Position: x - - - - - - - x
x - - - - - - - x
(a)
RS Code Symbols:
1 x 2 x 3 x 4 x 5 x 6 x 7 x 8
x - - - - - - - - - - - - - x
x - - - - - - - - - - - - - x
left most position
(b)
right most position

Figure 2: Illustration of a target RS codeword. Target codeword symbols are numbered; symbols from adjacent interleaved codewords indicated by x's. (a) Two positions of a TC error burst of length $\ell^*(2)+1$. The first position corrupts 3 RS code symbols of the target RS codeword, while the second position corrupts only 2 RS code symbols. (b) Left most and right most positions of a TC error burst of length $\ell > \ell^*(3)$ that will corrupt sequences of at least 3 RS code symbols.

code symbols depending on its position. This is illustrated in Figure 2a. In particular, when $\ell(\mathbf{e}) \le \ell^*(t_{RS})$ the target RS codeword is correctly decoded (so $n_e(\ell; n_{RS}, t_{RS}) = 0$).

Consider a TC error burst of length $\ell = \ell(\mathbf{e})$ in the range $\ell^*(t_{RS}) < \ell \le \ell^*(t_{RS}+1)$. As illustrated in Figure 2a, the number of positions of the TC error burst that will cause a specific RS error burst of $t_{RS}+1$ code symbols is $\ell - \ell^*(t_{RS})$. The number of positions of the RS error burst within the target RS codeword is $n_{RS} - t_{RS}$. Thus, $n_e(\ell; n_{RS}, t_{RS})$ $= (n_{RS} - t_{RS})\,(\ell - \ell^*(t_{RS}))$ for $\ell^*(t_{RS}) < \ell \le \ell^*(t_{RS}+1)$. Define

$$\tilde{n}_e(\ell; n, t) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{for } \ell \le \ell^*(t) \\ (n-t)(\ell - \ell^*(t)) & \text{for } \ell > \ell^*(t) \end{cases}. \tag{6}$$

Then we have shown that $n_e(\ell; n_{RS}, t_{RS}) = \tilde{n}_e(\ell; n_{RS}, t_{RS})$ for $\ell \le \ell^*(t_{RS}+1)$.

The situation for lengths $\ell > \ell^*(t_{RS}+1)$ is illustrated in Figure 2(b), which shows the leftmost and rightmost positions of the RS error burst that will cause an RS decoding error. All positions in between cause an error. After a little algebra, the result is reduced to

$$n_e(\ell; n_{RS}, t_{RS}) \;=\; \tilde{n}_e(\ell; n_{RS}, t_{RS}) \;-\; \tilde{n}_e(\ell; n_{RS}, t_{RS}+1). \tag{7}$$

This result is consistent with the previous result for $\ell \le \ell^*(t_{RS}+1)$, hence, (7) is valid for all lengths ℓ.

Now, to eliminate $n_e(\ell; n_{RS}, t_{RS})$, plug (6) and (7) into (5). The yields a bound that is the difference of two power series. From (6), the first power series sums over $\ell > \ell^*(t_{RS})$ and the second sums over $\ell > \ell^*(t_{RS}+1)$. Define the generating function

$$T_1(D; \ell^*) \stackrel{\text{def}}{=} \sum_{\ell=\ell^*+1}^{\infty} \sum_{d} (\ell - \ell^*)\, c(\ell, d)\, D^d. \tag{8}$$

Then the generating function bound is

$$P_{e1}(n_{RS}, t_{RS}) \leq C(t_{RS}) \left\{ (n_{RS} - t_{RS})\, T_1(D^*; \ell^*(t_{RS})) - (n_{RS} - t_{RS} - 1)\, T_1(D^*; \ell^*(t_{RS} + 1)) \right\}. \quad (9)$$

Our goal now is to obtain formulas evaluating the generating function $T_1(D; \ell^*)$ that appears in (9). As discussed in Section 2, in the standard theory the generating function is obtained as the transfer function of an error event enumeration diagram. The problem with the standard diagram is that it counts all error event error sequences $\mathbf{e}$. Here we only want to count the error events that produce TC error bursts of length $\ell > \ell^* = \ell^*(t)$, and the information weights in (8) are $(\ell - \ell^*(t))\, c(\ell, d)$.

The solution is to modify the error event enumeration diagram to eliminate paths of length $\ell(\mathbf{e}) \leq \ell^*$, as illustrated in Figure 3 for the example code of Figure 1. Nodes in this diagram are organized in columns, called stages. Successive transitions from stage ℓ to $\ell + 1$ replicate the structure of the original state diagram. There are three distinct phases in this the diagram. The first phase consists of just the initial branches from state 0 into stage 1 nodes. These branches must have non-zero TC information symbol error labels ($e_1 \neq 0$), and no 0 to 0 transitions are allowed. The second phase is the "TC error burst phase." This phase is essentially just the trellis diagram with state 0 nodes omitted. There are $\ell^* - 1$ stages in the error burst phase. Stage ℓ^* is the only stage that can loop back on itself. Finally, the "absorption phase" consists of all paths from stage ℓ^* into the absorbing state 0. These paths must satisfy the following two criteria: 1) the first branch of an absorption phase path (out of stage ℓ^*) is always an TC information symbol error, and 2) all subsequent branches cause no TC information symbol errors. Thus, the first branch of the absorption phase is actually the last information symbol error of the TC error burst. We thus conclude that this diagram enumerates all error paths $(0, \mathbf{e})$ of length $\ell(\mathbf{e}) > \ell^*$. The actual length $\ell(\mathbf{e})$ is $\ell^* + 1$ plus the number of state ℓ^* to ℓ^* transitions.

Branches in the error event enumeration diagram are labeled with the weight profiles. Recall that the uncoded bit of the Figure 1 example code gives rise to parallel branches. The possible sums of weight profiles for the parallel branches are $\psi_a(D) \stackrel{\text{def}}{=} D^1 + D^3 = \psi(D; 001) + \psi(D; 101) = \psi(D; 011) + \psi(D; 111)$, $\psi_b(D) \stackrel{\text{def}}{=} 1 + D^4 = \psi(D; 001) + \psi(D; 101)$, and $\psi_c(D) \stackrel{\text{def}}{=} 2D^2 = \psi(D; 010) + \psi(D; 110)$. In Figure 3, parallel branches are represented by single bold branches labeled with $\psi_a(D)$, $\psi_b(D)$ or $\psi_c(D)$. Some of the branches in the absorption phase, however, are single branches. For example, the branch from state 2 of stage ℓ^* into the absorbing state is the single branch with TC information symbol error label $e = 10$. The parallel branch with TC information label $e = 00$ is not a legitimate first branch in the absorption phase. These single branches are indicated as thin lines in Figure 3 with the explicit labels $\psi(D; \tilde{e})$.

In addition to the weight profiles, the initial branches, and all stage ℓ^* to stage ℓ^* transitions include a factor Z. These Z factors enumerate the $\ell - \ell^*$ factor of the information weight $(\ell - \ell^*(t))c(\ell, d)$.

Let $T(Z, D; \ell^*)$ denote the "transfer function" of the error event enumeration diagram

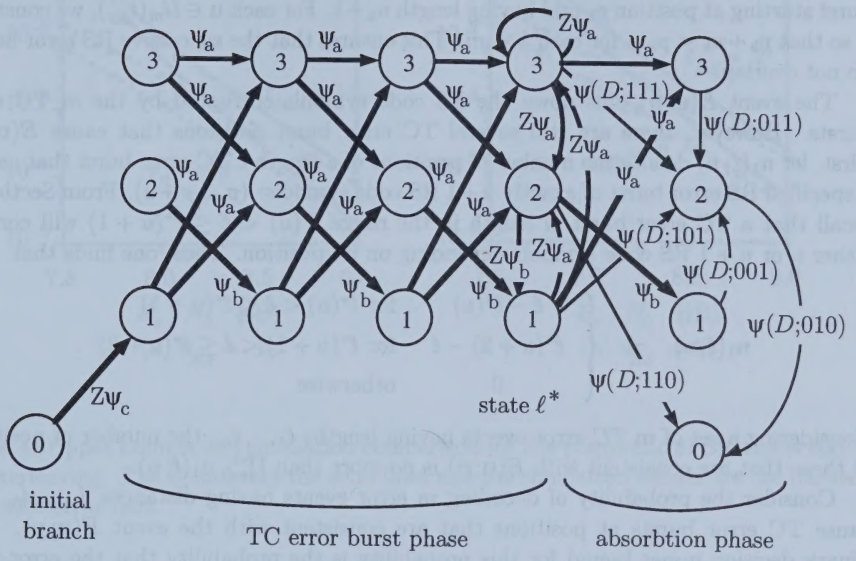


Figure 3: The error event enumeration diagram for for TC error bursts of length $\ell > \ell^* = 4$.

of Figure 3. Then, the desired generating function is

$$T_1(D;\ell^*) \;=\; \frac{\partial}{\partial Z}\, T(Z,D;\ell^*)\,\Big|_{Z=1}. \tag{10}$$

Following the standard transfer function theory for diagrams such as that of Figure 3, we find that the transfer function may, in general, be evaluated by the formula

$$T(Z,D;\ell^*) \;=\; Z\,\mathbf{c}(D)^t\,\mathbf{A}(D)^{\ell^*-1}\,[\mathbf{I} - Z\,\mathbf{A}(D)\,]^{-1}\,\mathbf{b}(D) \tag{11}$$

where $A_{s,\sigma}(D) \stackrel{\text{def}}{=} \sum_{\widetilde{e}\,\in\,E(\sigma\to s)} \psi(D;\widetilde{e})$ with $E(\sigma \to s)$ being the set of branch labels for trellis branches that transition from state σ into state s, $b_s(D) \stackrel{\text{def}}{=} \sum_{\widetilde{e}\,\in\,E(0\to s)} \psi(D;\widetilde{e})$, and $c_s(D) \stackrel{\text{def}}{=} \sum_{\mathbf{e}\,\in\,F(s)} \Psi(D;\mathbf{e})$ where $F(s)$ is the set of absorption phase paths starting in state s at stage ℓ^*.

4. MULTIPLE ERROR EVENTS

We now expand the analysis to get bounds for $P_{em}(n_{RS}, t_{RS}) \stackrel{\text{def}}{=}$ the probability of an RS decoding error caused by m TC error bursts. Limited space does not allow a complete exposition of the derivation; we sketch the main points.

For $m \geq 2$, let $\mathcal{U}_m(t_{RS})$ denote the set of all m-vectors of non-negative integers $\mathbf{u} = (u_1, .., u_m)$ such that $\sum_{k=1}^{m}(u_k+1) > t_{RS}$ and $\sum_{k\neq j}(u_k+1) \leq t_{RS}$ for each $j = 1, .., m$. Let $E(\mathbf{u};\mathbf{p})$, where $\mathbf{p} = (p_1, .., p_m)$, be the event that m error events occur within the

scope of the target RS codeword such that the kth TC error burst causes an RS error burst starting at position p_k and having length u_k+1. For each $\mathbf{u} \in \mathcal{U}_m(t_{RS})$, we constrain $\mathbf{p}$ so that $p_k + u_k < p_{k+1}$ for each $k < m$. This ensures that the successive RS error bursts do not overlap.

The event $E(\mathbf{u};\mathbf{p})$ pins down the RS code symbols corrupted by the m TC error bursts. However, there are still several TC error burst positions that cause $E(\mathbf{u};\mathbf{p})$. First, let $n_1(\ell;u)$ denote the number of positions of a length ℓ TC error burst that causes a specified RS error burst of exactly $u+1$ RS code symbols: $(p,..,p+u)$. From Section 3, recall that a TC error burst of length in the range $\ell^*(u) < \ell \le \ell^*(u+1)$ will corrupt either u or $u+1$ RS code symbols depending on its position. Thus, one finds that

$$n_1(\ell;u) \;=\; \begin{cases} \ell - \ell^*(u) & \text{for } \ell^*(u) < \ell \le \ell^*(u+1) \\ \ell^*(u+2) - \ell & \text{for } \ell^*(u+1) < \ell \le \ell^*(u+2) \\ 0 & \text{otherwise} \end{cases} . \tag{12}$$

Considering a set of m TC error events having lengths $\ell_1,..,\ell_m$, the number of positions of these that are consistent with $E(\mathbf{u};\mathbf{p})$ is no more than $\prod_{i=1}^{m} n_1(\ell;u)$.

Consider the probability of decoding m error events having distances $d_1,..,d_m$ that cause TC error bursts at positions that are consistent with the event $E(\mathbf{u};\mathbf{p})$. The binary decision upper bound for this probability is the probability that the error path metric exceed the correct path metric sum independently for each of the m error events. Thus, the probability of decoding m distinct error events having distances $d_1,..,d_m$ is $\prod_{k=1}^{m} Q\left(\sqrt{d_k\,\mathcal{E}_s/2N_0}\right) \le \prod_{k=1}^{m} C(u_k)\,(D^*)^{d_k}$.

The upper bounds of the last two paragraphs both have product form. Recalling $c(\ell,d) \stackrel{\text{def}}{=} \sum_{\mathbf{e}:\,\ell(\mathbf{e})=\ell} a(d;\mathbf{e})$, we also have

$$\sum_{\mathbf{e}_1,..,\mathbf{e}_m:\,\ell(\mathbf{e}_k)=\ell_k} \prod_{k=1}^{m} a(d_k;\mathbf{e}_k) \;=\; \prod_{k=1}^{m} \sum_{\mathbf{e}:\,\ell(\mathbf{e})=\ell_k} a(d_k;\mathbf{e}) \;=\; \prod_{k=1}^{m} c(\ell_k,d_k).$$

Using these product form expressions, we obtain

$$\mathcal{P}\big(E(\mathbf{u};\mathbf{p})\big) \;\le\; \prod_{k=1}^{m} C(u_k)\,V(D^*;u_k)$$

where $V(D;u) \stackrel{\text{def}}{=} \sum_{\ell,d} n_1(\ell;u)\,c(\ell,d)\,D^d$. The above upper bound for $\mathcal{P}(E(\mathbf{u};\mathbf{p}))$ does not depend on the position vector $\mathbf{p}$. The number of admissible position vectors $\mathbf{p}$ is $\binom{n-\sigma(\mathbf{u})}{m}$ where $\sigma(\mathbf{u}) \stackrel{\text{def}}{=} \sum_{k=1}^{m} u_k$. Thus, summing the $\mathcal{P}(E(\mathbf{u};\mathbf{p}))$ bound over admissible $\mathbf{u}$ and $\mathbf{p}$, the union bound is

$$P_{em}(n_{RS},t_{RS}) \;\le\; \sum_{\mathbf{u}\in\mathcal{U}_m(t_{RS})} \binom{n-\sigma(\mathbf{u})}{m} \prod_{k=1}^{m} C(u_k)\,V(D^*;u_k). \tag{13}$$

At this point we need only determine formulas for evaluating $V(D;u)$ that can be applied to (13). The final result is obtained by applying (12) to the definition of $V(D;u)$. There are two finite ranges of values of ℓ in (12). We realize these finite sums by differencing power series. The result is

$$V(D;u) \;=\; T_1(D;\ell^*(u)) \;-\; 2\,T_1(D;\ell^*(u+1)) \;+\; T_1(D;\ell^*(u+2)) \tag{14}$$

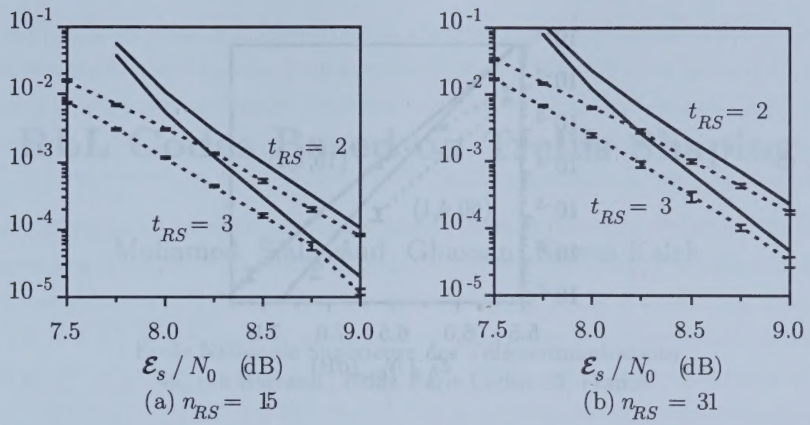


Figure 4: Upper bounds and simulation comparison for the pragmatic 8PSK trellis code. No interleaving. The bounds are the solid lines and the simulation results are the dashed lines with error bars.

for $u \geq 1$, and

$$V(D;0) = r_s \left[\zeta(1,D) + T_0(D;1)\right] + T_1(D;1) - 2T_1(D;\ell^*(1)) + T_1(D;\ell^*(2)). \quad (15)$$

Let $F(0)$ denote the set of error event error sequences $\mathbf{e}$ such that $e_1 \neq 0$ and $e_k = 0$ for all $k > 1$. Then $\zeta(1,D) = \sum_{\mathbf{e} \in F(0)} \Psi(D;\mathbf{e})$. $F(0)$ will contain all single branch state 0 to state 0 paths that cause TC information symbol errors.

5. NUMERICAL EXAMPLES AND CONCLUDING REMARKS

In order to demonstrate bound convergence of the bound for practical operating points, Figure 4 compares the bound to simulation results. The trellis code was the 64 state 8PSK pragmatic trellis code described in [3]. The Reed-Solomon code parameters are $n_{RS} = 15$ and 31 with $t_{RS} = 2$ and 3, with no interleaving.

It should be noted that these concatenated coding systems were extremely difficult to simulate. The data points for $\mathcal{E}_s/N_0 = 9$ dB in Figure 4, for example, required four days of continuous running time on a SPARC 10 workstation. This high computational cost limited our ability to simulate to systems with only moderately powerful coding systems.

We also used simulations to investigate the bound tightness with interleaving. A numerical example is illustrated in Figure 5. Due the difficulty of simulating the more powerful systems (as noted above), this comparison utilized the simple 4 state code of Figure 1. The outer code and interleaver parameters are (n_{RS}, t_{RS}, d_I) = (15,1,4) and (60,4,1). The reason for this choice of parameters is that we also wished to investigate the trade-off between using a less complex Reed-Solomon code with interleaving vs. using a more powerful Reed-Solomon code. Both the systems have the same outer code rate: $R_{RS} = (n_{RS} - 2t_{RS})/n_{RS} = 13/15$.

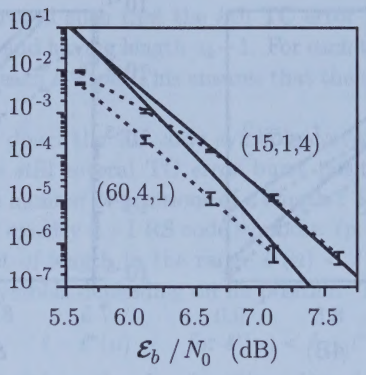


Figure 5: Upper bounds and simulation comparison for the 4 state 8PSK trellis code of Figure 1. The bounds are the solid lines and the simulation results are the dashed lines with error bars. The outer code and interleaver parameters are $(n_{RS}, t_{RS}, d_I) = (15,1,4)$ and (60,4,1).

Acknowledgment

The author would like to express his sincere appreciation to Fairchild Data Corporation for financial support, and in particular, to Terry Richmond of Fairchild for many fruitful and informative discussion. The author is also indebted to Chamath Abhayagunawardhana for his aid with the numerical work.

References

[1] M. Rouanne and D. J. Costello, Jr., "An algorithm for computing the distance spectrum of trellis codes," *IEEE J. Select. Areas Commun.*, **JSAC-7**, pp. 929-940, 1989.

[2] G. Ungerboeck, "Trellis-coded modulation with redundant signal sets, Part I: introduction, Part II: state of the art," *IEEE Commun. Mag.*, **25**, pp. 5-21, 1987.

[3] A. J. Viterbi, J. K. Wolf, E. Zehavi and R. Padovani, "A pragmatic approach to trellis-coded modulation," *IEEE Commun. Mag.*, **27**, pp. 11-19, 1989.

[4] A. J. Viterbi and J. K. Omura, *Principles of Digital Communications and Coding*. McGraw-Hill, New York, 1979.

[5] E. Zehavi and J. K. Wolf, "On the performance evaluation of trellis codes," *IEEE Trans. Inform. Theory*, **IT-33**, pp. 196-202, 1987.

RLL Codes Based on Trellis Shaping

Mohamed Siala and Ghassan Kawas Kaleh

Ecole Nationale Supérieure des Télécommunications
46, rue Barrault, 75634 Paris Cedex 13, France.

Abstract — A (d,k) phrase is a sequence of i successive zeros followed by a single one, $d \leq i \leq k$. A (d,k) runlength-limited code is the concatenation of such phrases. To approach the Shannon capacity, short phrases should be used more frequently than long ones. We use the trellis shaping technique, proposed to obtain shaping gain in coded modulation systems, as a method for guaranteeing this behavior. We describe the encoding and decoding method and show that for certain configurations the proposed code can achieve the Shannon capacity of the (d,k) constraint.

1 Introduction

Runlength-limited (RLL) codes, also known as (d,k) RLL codes, are typically used in digital magnetic recording systems to increase the recording density and control self clocking. They also have a potential application in optical transmission by soliton pulses because they guarantee the necessary condition that the pulses do not overlap.

Denoting by 0^m a sequence of m successive zeros, we define for a (d,k) RLL code the following alphabet:

$$\mathcal{H}_{dk} = \{\text{“}0^d1\text{”}, \text{“}0^{d+1}1\text{”}, \ldots, \text{“}0^k1\text{”}\}.$$

. Each element in $\mathcal{H}_{dk}$, called a phrase, is a variable-length sequence of i consecutive symbols “0”, $d \leq i \leq k$, followed by a single symbol “1”. A (d,k)-sequence is the concatenation of phrases drawn from $\mathcal{H}_{dk}$. Phrases can be represented by their lengths in binary symbols. Let $\mathcal{J} = \{d+1, d+2, \ldots, k+1\}$ be the set of phrase lengths, then there is a one-to-one mapping $l(.)$ between $\mathcal{H}_{dk}$ and $\mathcal{J}$, where the image of the phrase “0^e1” is $l(\text{“}0^e1\text{”}) = e+1$. The length of any sequence in $(\mathcal{H}_{dk})^N$ is defined as the sum of the lengths of all its components. We also define $F_0(.)$ as the mapping from $(\mathcal{H}_{dk})^N$ into $\mathcal{J}^N$ which maps a vector of N phrases into its length.

A (d,k) RLL encoder maps unconstrained binary sequences, referred to as information binary sequences, into (d,k) RLL constrained sequences. Encoding methods are described in [5]. Here, we present a new approach for constructing variable length (d,k) RLL codes based on trellis shaping that Forney [3] has designed to obtain shaping gain, in addition to coding gain, in coded modulation systems.

Trellis shaping is a generalization to infinite dimensions of Voronoi shaping [2]. For a finite dimensional constellation used on the Gaussian channel, the Voronoi shaping region is a good approximation of the optimal shaping region. For an N-dimensional lattice code Λ and for a given number of codewords, one considers only those points of Λ that are the closest to the origin in Euclidean distance or its related norm $||.||_2$. In practice the optimal shaping region, which in this case is a sphere, is approximated by the Voronoi region of a sublattice Λ_s of Λ. This approximation becomes tighter when the dimension N becomes large.

When considering RLL codes, we work with the l_1-distance or its corresponding one norm, denoted by $||.||_1$, and the optimal region we would like to approximate has not a spherical form, as in coded modulation for the Gaussian channel, but a pyramidal form. Moreover, this pyramidal form is not centered in the origin zero. Consequently, a direct use of Voronoi shaping will not lead to a high coding rate. Subsequently we propose to use a simple mapping that tries to transform the optimal pyramidal region in a given dimension into a region which can be efficiently approximated by a Voronoi region centered in the origin of the Euclidean space. The resulting transformation is a generalization of the trivial mapping $F_0(.)$, defined previously. This mapping, denoted by $F(.)$, operates therefore from $(\mathcal{H}_{dk})^N$ into $\boldsymbol{Z}^N$.

The mapping $F(.)$ is used as an intermediate step in labeling the shaping trellis branches by finite length sequences in $(\mathcal{H}_{dk})^N$. The labeling process is done in two steps. First, we work during the trellis shaping design with the points of $\mathcal{R}_N = F((\mathcal{H}_d^k)^N)$, instead of their corresponding finite sequences in $(\mathcal{H}_{dk})^N$. Secondly, we proceed to replace trellis branches labels in $\mathcal{R}_N$ by their counterparts in $(\mathcal{H}_{dk})^N$. This approach is more efficient than working directly with $(\mathcal{H}_{dk})^N$.

Therefore, the set $\mathcal{R}_N$ is defined as

$$\mathcal{R}_N = \{m(u,z) : u \in (\boldsymbol{Z}_2)^{n_u}, z \in (\boldsymbol{Z}_2)^{n_s}\}.$$

It is partitioned into disjoint subsets $\{\mathcal{A}_N(z), z \in (\boldsymbol{Z}_2)^{n_s}\}$. Also, it is partitioned into other disjoint subsets $\{\mathcal{B}_N(u), u \in (\boldsymbol{Z}_2)^{n_u}\}$, where

$$\mathcal{A}_N(z) = \{m(u,z), u \in (\boldsymbol{Z}_2)^{n_u}\}\ ,\quad \mathcal{B}_N(u) = \{m(u,z), z \in (\boldsymbol{Z}_2)^{n_s}\}.$$

2 RLL Encoding

According to [3], trellis shaping is based on:

1. a binary rate-k_s/n_s convolutional shaping code $\mathcal{C}_s$,
2. a partition of the region $\mathcal{R}_N$ into 2^{n_s} subregions $\{\mathcal{A}_N(z), z \in (\boldsymbol{Z}_2)^{n_s}\}$, where $\boldsymbol{Z}_2 = \boldsymbol{Z}/2\boldsymbol{Z}$,
3. a shaping decoder.

The elements of RLL encoder are illustrated in Figure 1. The input sequence is binary and divided into two parts (u_j, s_j). The first part, u_j, is an uncoded n_u-tuple. The second part, a syndrome r_s-tuple s_j, is an input to an r_s-input, n_s-output coset representative generator $(\boldsymbol{H}_s^{-1})^t$ for a rate-k_s/n_s convolutional shaping code $\mathcal{C}_s$, where $k_s = n_s - r_s$.

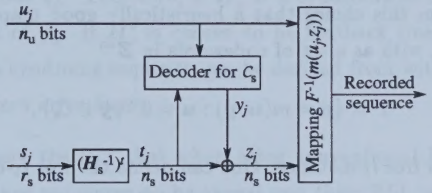


Figure 1: RLL Encoder.

According to [3], trellis shaping is based on:

1. a binary rate-k_s/n_s convolutional shaping code $\mathcal{C}_s$,
2. a partition of the region $\mathcal{R}_N$ into 2^{n_s} subregions $\{\mathcal{A}_N(z), z \in (\boldsymbol{Z}_2)^{n_s}\}$, where $\boldsymbol{Z}_2 = \boldsymbol{Z}/2\boldsymbol{Z}$,
3. a shaping decoder.

From the syndrome sequence $\boldsymbol{s} = (\ldots, s_{i-1}, s_i, s_{i+1}, \ldots)$, the coset representative generator $(\boldsymbol{H}_s^{-1})^t$ generates a coset representative sequence $\boldsymbol{t} = \boldsymbol{s}(\boldsymbol{H}_s^{-1})^t$ of coset representative n_s-tuples t_j. The shaping decoder, using the two sequences $(\boldsymbol{u}, \boldsymbol{t})$ to make its decision, selects a shaping code sequence $\boldsymbol{y}_s \in \mathcal{C}_s$ and adds it to $\boldsymbol{t}$ to form a shaping sequence $\boldsymbol{z} = \boldsymbol{y}_s \oplus \boldsymbol{t}$, where $\oplus$ represents term by term addition modulo 2. The resulting shaping sequence $\boldsymbol{z}$ falls therefore in the same coset $\mathcal{C}_s \oplus \boldsymbol{t}$ of $\mathcal{C}_s$ as the initial sequence $\boldsymbol{t}$.

Each couple (u_j, z_j) is mapped into a point $p_j = m(u_j, z_j)$ in $\mathcal{R}_N$ by a one-to-one mapping function $m(.,.)$. For a fixed value of $u \in (\boldsymbol{Z}_2)^{n_u}$, $m(u, z)$ is in different subregions of $\mathcal{R}_N$ for different values of $z \in (\boldsymbol{Z}_2)^{n_s}$.

The N-dimensional region $\mathcal{R}_N$ should therefore have exactly 2^n points where $n = n_u + n_s$. The number of input bits per N dimensions is $n_u + r_s$. The shaping operation adds $k_s = n_s - r_s$ bits per N dimensions. Without shaping the number of input bits per N dimensions should be equal to n. The shaping operation reduces this number by k_s bits per N dimensions. As in [3], we call the parameter $\kappa = k_s/N$ the normalized informativity of the shaping code $\mathcal{C}_s$, in bits per dimension.

As we have done for the length operator, we generalize here the definition of the operators $m(.,.)$ and $F(.)$ to any arbitrary cross product of their starting or destination sets. Intuitively, an optimal mapping $F(.)$ on $(\mathcal{H}_{dk})^N$ should associate any RLL sequence in $(\mathcal{H}_{dk})^N$ with a point in $\mathcal{R}_N$ whose l_1-norm is equal to the length of this sequence. Moreover, the image of $(\mathcal{H}_{dk})^N$ by this optimal mapping should be symmetric around the origin of $\boldsymbol{Z}^N$. It follows from this choice that a heuristically good mapping $m(.,.)$ is one that provides a trellis code, with as a set of codewords in $\boldsymbol{Z}^\infty$

$$\mathcal{T} = \{\boldsymbol{p} = m(\boldsymbol{u}, \boldsymbol{y}) : \boldsymbol{u} = 0^\infty, \boldsymbol{y} \in \mathcal{C}_s\}, \tag{1}$$

that has the maximum free l_1-distance. One can define the free l_1-distance of a trellis code as the smallest l_1-distance between signals along differing paths in its trellis.

The set

$$\mathcal{S}(\boldsymbol{u}, \boldsymbol{t}) = \left\{\boldsymbol{w} = F^{-1} \circ m(\boldsymbol{u}, \boldsymbol{z}) : \boldsymbol{z} \in \mathcal{C}_s + \boldsymbol{t}\right\}$$

is regarded as an equivalent class of possible RLL sequences, any one of which represents the input bit sequence $(\boldsymbol{u}, \boldsymbol{s})$ that generates $(\boldsymbol{u}, \boldsymbol{t})$. The criterion of the shaping decoder is to minimize the average length per recorded bit, i.e., given $\mathcal{S}(\boldsymbol{u}, \boldsymbol{t})$, its goal is to choose the sequence $\boldsymbol{w} \in \mathcal{S}(\boldsymbol{u}, \boldsymbol{t})$ of least length $l(\boldsymbol{w})$.

The set of all sequences $\boldsymbol{w} \in \mathcal{S}(\boldsymbol{u}, \boldsymbol{t})$ may be represented by the same trellis diagram as that of the code $\mathcal{S}_s$, except that a branch labeled by $y_{s,j}$ in the trellis diagram for $\mathcal{C}_s$ is instead labeled by the corresponding basic sequence in $\mathcal{H}_{dk}$

$$w_j = F^{-1} \circ m(u_j, z_j), \quad z_j = y_{s,j} \oplus t_j.$$

The set of all $\boldsymbol{w} \in \mathcal{S}(\boldsymbol{u}, \boldsymbol{t})$ corresponds precisely to the set of all possible paths through the relabeled trellis diagram. If the weight assigned to a branch in the trellis with label w_j is proportional to a shifted version of $l(w_j)$, then the Viterbi algorithm can be used to find

the minimum length sequence $\boldsymbol{w} \in \mathcal{S}(\boldsymbol{u}, \boldsymbol{t})$. Since the trellis structure is the same as that of the trellis diagram for $\mathcal{C}_s$, the complexity of such a search is the same as that of a decoder for $\mathcal{C}_s$, and the shaping decoder may be regarded as a decoder for $\mathcal{C}_s$.

As we have said previously, to achieve high RLL code rates, the mapping $m(.,.)$ should be chosen such that the minimum l_1-norm between any two sequences generated by the same trellis diagram as that of $\mathcal{C}_s$, with the branches labeled by $m(u_j, y_{s,j})$ instead of $y_{s,j}$, should be maximized for arbitrary values of u_j. The achievable RLL coding rate depends therefore strongly on the mapping $F(.)$.

3 RLL Decoding

If we assume that the RLL sequence contains no shifts or insertion errors, then we can deduce without errors the corresponding shaping sequence $\boldsymbol{z} = \boldsymbol{y}_s + \boldsymbol{t}$ that lies in the coset $\mathcal{C}_s \oplus \boldsymbol{t}$ of the shaping code $\mathcal{C}_s$. Therefore, if $\boldsymbol{H}_s^t$ is a syndrome-former matrix for $\mathcal{C}_s$, then $\boldsymbol{z}\boldsymbol{H}_s^t = \boldsymbol{s}$, independent of $\boldsymbol{y}_s$. If $\boldsymbol{H}_s^t$ is chosen to be feedback free, as is always possible, then an estimate of the syndrome sequence can be derived from without catastrophic error propagation, even if is not error-free.

We should stress here the fact that when using conventional RLL coding, the correspondence between binary sequences to be stored and their RLL sequences are error-free. However, due to shift and insertion errors, we cannot avoid having erroneous data even when using sliding block decoding. In our encoding scheme, errors are not only due to insertions and shifts but also due to the finite size of decoding window when using the Viterbi algorithm. Therefore the window size should be chosen such that the bit error rate due to shifts and insertions be comparable to that introduced by finite size decoding window. In other words, it is not necessary to have a complicated errorless RLL coding scheme when we cannot avoid having errors due to shifts or insertions.

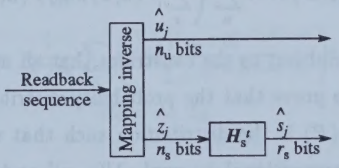


Figure 2: Decoder for the RLL coding system with trellis shaping.

In summary, the operation of the receiver is illustrated in Figure 2. An RLL sequence $\hat{\boldsymbol{w}}$ is decomposed into its corresponding two-part label $(\hat{\boldsymbol{u}}, \hat{\boldsymbol{z}})$. The estimated recorded binary sequences $(\hat{\boldsymbol{u}}, \hat{\boldsymbol{s}})$ are recovered from the map $\hat{\boldsymbol{s}} = \hat{\boldsymbol{z}}\boldsymbol{H}_s^t$. Here we stress the fact that when there are no shift or insertion errors, the errors due to the decoding window finite size result in errors in the binary sequence $\hat{\boldsymbol{s}}$ only.

4 Bounds on achievable rates

Following a similar development as in [4] and [3], we give bounds for the best possible achievable RLL coding rates when using any mapping $F(.)$ and $m(.,.)$ and any shaping trellis. These achievable rates are expressed as a function of the normalized informativity κ. Let us label each point $m(u,z)$ of the subset $\mathcal{R}_N = F((\mathcal{H}_{dk})^N)$ by the two-part label (u,z). With random input data, all possibilities for the one part u will be equally likely. The shaping operation does not affect this part of the label, so the 2^{n_u} subsets $\mathcal{B}_N(u) = \{m(u,z) : z \in (\boldsymbol{Z}_2)^{n_s}\}$ of $\mathcal{R}_N$ will be equiprobable.

For notational convenience, we denote by $l(u,z)$ the length of the finite sequence in $(\mathcal{H}_{dk})^N$ which has $m(u,z)$ as image by $F(.)$, i.e., $l(u,z) = l(F^{-1}(m(u,z)))$.

The effect of the shaping operation is to modify the probabilities of the 2^{n_s} points within any of these subsets $\mathcal{B}_N(u)$, so as to make the lower-length sequences in $\mathcal{H}_d^k$ more probable and the higher-length sequences less probable, thus achieving an overall probability distribution $P(u,z)$ over the signal points in $\mathcal{R}_N$, or equivalently over the finite length sequences in $(\mathcal{H}_{dk})^N$. The average sequence length per N dimensions is given by

$$L_N(P) = \sum_u \left(\sum_z P(u,z) l(u,z) \right),$$

and the entropy of the N-dimensional distribution $P(u,z)$ is

$$H_N(P) = -\sum_u \left(\sum_z P(u,z) \log_2 P(u,z) \right),$$

in bits per N dimensions. Subject to the constraint that all marginal probabilities $P(u)$ be equal to 2^{-n_u}, it is easy to prove that the probability distribution $P(u,z)$ that minimizes $L_N(P)$ for a constant $H_N(P)$ is the distribution such that within any subset $\mathcal{B}_N(u)$ the probabilities $P(u,z)$ are proportional to $\exp[-\lambda l(u,z)]$ and sum to 2^{-n_u}, where λ is a Lagrange multiplier.

$$P(u,z) = P_\lambda = 2^{-n_u} \frac{2^{-\lambda l(u,z)}}{\sum_{z'} 2^{-\lambda l(u,z')}}. \tag{2}$$

This gives a parametric optimum trade-off between $L_N(P_\lambda)$ and $H_N(P_\lambda)$. The normalized informativity corresponding to λ is then given by $\kappa(\lambda) = (n_u + n_s - H_N(\lambda))/N$

More specifically, the average length and entropy per dimension (or per phrase) are given respectively by

$$L(P_\lambda) = \frac{1}{N 2^{n_u}} \sum_u \left(\frac{\sum_z l(u,z) 2^{-\lambda l(u,z)}}{\sum_z 2^{-\lambda l(u,z)}} \right), \tag{3}$$

and

$$H(P_\lambda) = \frac{1}{N 2^{n_u}} \sum_u \log_2 \left(2^{n_u} \sum_z 2^{-\lambda l(u,z)} \right) + \lambda L(P_\lambda). \tag{4}$$

Given $L(P_\lambda)$ and $H(P_\lambda)$ we can define the RLL achievable code rate and the normalized informativity respectively as

$$\rho(\lambda) = \frac{H(\lambda)}{L(\lambda)}, \tag{5}$$

and

$$\kappa(\lambda) = \frac{n_u + n_s}{N} - H(P_\lambda). \tag{6}$$

Using the expressions of $L(P_\lambda)$ and $H(P_\lambda)$ given by (3) and (4) we obtain the following expression of the achievable RLL code rate

$$\rho(\lambda) = \lambda + \frac{\sum_u \log_2 \left(2^{n_u} \sum_z 2^{-\lambda l(u,z)} \right)}{\sum_u \left(\frac{\sum_z l(u,z) 2^{-\lambda l(u,z)}}{\sum_z 2^{-\lambda l(u,z)}} \right)}. \tag{7}$$

For a given mapping $F^{-1} \circ m(.,.)$, the maximum achievable rate ρ_0 is obtained by cancelling the derivative of (7). We find

$$\rho_0 = \lambda_0, \tag{8}$$

where λ_0 verifies

$$\prod_u \left(2^{n_u} \sum_z 2^{-\lambda_0 l(u,z)} \right) = 1. \tag{9}$$

In this case the length and entropy per dimension and the normalized informativity are given respectively by

$$L_0 = \frac{1}{N 2^{n_u}} \sum_u \left(\frac{\sum_z l(u,z) 2^{-\lambda_0 l(u,z)}}{\sum_z 2^{-\lambda_0 l(u,z)}} \right),$$

$$H_0 = \lambda_0 \, L_0$$

and

$$\kappa_0 = \frac{n_u + n_s}{N} - H_0.$$

To maximize $\rho_0 = \lambda_0$ we should choose $n_u = 0$. In this case the value of the maximum rate ρ_0 , denoted by $\mathrm{Cap}(d,k)$ is called the (d,k) RLL code capacity. Using (8) and (9), this capacity, as is well known (see [5]), is given by

$$\mathrm{Cap}(d,k) = \log_2(x_0),$$

where x_0 is the maximum solution of

$$\sum_{i=d+1}^{k+1} x^{-i} = 1,$$

For our needs in the next section, we should remember the fact that, for a given normalized informativity κ, the shaping code $\mathcal{C}_s$ rate k_s/n_s is such that $k_s = N\kappa$. Therefore, to achieve the best RLL code rate, the rate of the shaping code $\mathcal{C}_s$ should be chosen near $N\kappa_0/n_s$.

4.1 Numerical results

According to the expressions of (P_λ) and $H(P_\lambda)$ in (3) and (4) the characteristics of the subsets $\mathcal{A}_N(z), z \in (\boldsymbol{Z}_2)^{n_s}$ which determine the RLL coding quality can be completely specified by the subsets

$$\hat{\mathcal{A}}_N(z) = F_0 \circ F^{-1}\left(\mathcal{A}_N(z)\right), z \in (\boldsymbol{Z}_2)^{n_s},$$

which we can easily represent by a constellation. Similarly, the essential characteristics of the subsets $\mathcal{B}_N(u), u \in (\boldsymbol{Z}_2)^{n_u}$ are summarized in the subsets

$$\hat{\mathcal{B}}_N(u) = F_0 \circ F^{-1}\left(\mathcal{B}_N(u)\right), u \in (\boldsymbol{Z}_2)^{n_u}.$$

. As an example we consider the $(0,7)$ RLL code. The curve of $\rho(\lambda)$ as a function of $\kappa(\lambda)$ for this choice is plotted in Figure 3.

Numerical results giving the maximum achievable RLL coding rate ρ_0 and the optimum shaping code rate $N\kappa_s/n_s$ as a function of n_u, n_s and some of the characteristics of the sets $\mathcal{A}_N$, when $N = 1$, are given for these and other constraints in Table 1. Only two mappings $m(.,.)$ are considered in this table. The first one, proposed by Calderbank and Ozarow [1], and called here an "inverse" mapping, is such that the lengths of the elements of two adjacent subsets $\mathcal{A}_N(z)$ and $\mathcal{A}_N(z \pm 1)$ are alternatively increasing and decreasing functions of the parameter $u \in (\boldsymbol{Z}_2)^{n_u}$ indexing the subsets $\mathcal{B}_N(u)$. The second mapping, called here a "direct" mapping, is such that the lengths of the elements of all the subsets $\mathcal{A}_N(z)$, $z \in (\boldsymbol{Z}_2)^{n_s}$ are all either increasing or decreasing functions of the parameter u. According to the results in Table 1, "inverse" mapping theoretically outperforms "direct" mapping.

d/i	(d,k)	n_u	n_s	ρ_0	κ_0	$\frac{N\kappa_0}{n_s}$	d/i	(d,k)	n_u	n_s	ρ_0	κ_0	$\frac{N\kappa_0}{n_s}$
/	(0,1)	0	1	0.6942	0.0406	0.0406	d	(5,12)	1	2	0.3358	0.1789	0.0894
/	(1,2)	0	1	0.4057	0.0141	0.0141	d	(6,13)	1	2	0.3002	0.1455	0.0727
/	(0,3)	0	2	0.9468	0.3283	0.1641	d	(7,14)	1	2	0.2718	0.1207	0.0604
/	(1,4)	0	2	0.6174	0.1532	0.0766	i	(0,7)	1	2	0.9831	0.9494	0.4747
/	(2,5)	0	2	0.4650	0.0897	0.0448	i	(1,8)	1	2	0.6831	0.6045	0.3023
/	(3,6)	0	2	0.3746	0.0591	0.0295	i	(2,9)	1	2	0.5363	0.4195	0.2097
d	(0,3)	1	1	0.9073	0.2374	0.2374	i	(3,10)	1	2	0.4457	0.3089	0.1544
d	(1,4)	1	1	0.6069	0.1172	0.1172	i	(4,11)	1	2	0.3831	0.2373	0.1187
d	(2,5)	1	1	0.4605	0.0699	0.0699	i	(5,12)	1	2	0.3369	0.1882	0.0941
d	(3,6)	1	1	0.3723	0.0465	0.0465	i	(6,13)	1	2	0.3010	0.1531	0.0765
i	(0,3)	1	1	0.9336	0.2634	0.2634	i	(7,14)	1	2	0.2724	0.1270	0.0635
i	(1,4)	1	1	0.6157	0.1383	0.1383	d	(0,7)	2	1	0.8520	0.5767	0.5767
i	(2,5)	1	1	0.4645	0.0845	0.0845	d	(1,8)	2	1	0.6366	0.3998	0.3998
i	(3,6)	1	1	0.3744	0.0568	0.0568	d	(2,9)	2	1	0.5136	0.2902	0.2902
/	(0,7)	0	3	0.9971	1.0335	0.3445	d	(3,10)	2	1	0.4327	0.2193	0.2193
/	(1,8)	0	3	0.6853	0.6279	0.2093	d	(4,11)	2	1	0.3749	0.1714	0.1714
/	(2,9)	0	3	0.5369	0.4289	0.1430	d	(5,12)	2	1	0.3313	0.1375	0.1375
/	(3,10)	0	3	0.4460	0.3135	0.1045	d	(6,13)	2	1	0.2970	0.1127	0.1127
/	(4,11)	0	3	0.3833	0.2399	0.0800	d	(7,14)	2	1	0.2694	0.0940	0.0940
/	(5,12)	0	3	0.3369	0.1898	0.0633	i	(0,7)	2	1	0.8915	0.5337	0.5337
/	(6,13)	0	3	0.3011	0.1541	0.0514	i	(1,8)	2	1	0.6594	0.4068	0.4068
/	(7,14)	0	3	0.2724	0.1276	0.0425	i	(2,9)	2	1	0.5277	0.3149	0.3149
d	(0,7)	1	2	0.9580	0.9102	0.4551	i	(3,10)	2	1	0.4419	0.2487	0.2487
d	(1,8)	1	2	0.6740	0.5743	0.2872	i	(4,11)	2	1	0.3812	0.2004	0.2004
d	(2,9)	1	2	0.5318	0.3982	0.1991	i	(5,12)	2	1	0.3358	0.1643	0.1643
d	(3,10)	1	2	0.4432	0.2933	0.1466	i	(6,13)	2	1	0.3004	0.1369	0.1369
d	(4,11)	1	2	0.3815	0.2254	0.1127	i	(7,14)	2	1	0.2720	0.1157	0.1157

Table 1: Numerical results for several (d,k) RLL codes. (d/i stands for "direct" or "inverse" mapping)

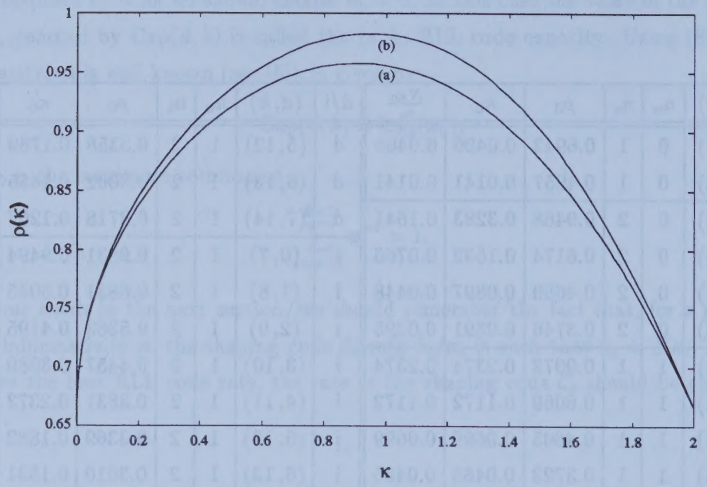


Figure 3: The best achievable (0,7) RLL rate ρ as a function of the normalized informativity κ when $n_u = 1$. (a): Direct mapping. (b): Inverse mapping.

References

[1] A. R. Calderbank and L. H. Ozarow, "Nonequiprobable Signaling on the Gaussian Channel," *IEEE Trans. Inform. Theory*, Vol. 36, NO. 4, July 1990.

[2] J. H. Conway and N. J. A. Sloane, "A Fast Encoding Method for Lattice Codes and Quantizers," *IEEE Trans. Inform. Theory*, Vol IT-29, pp. 820-824, Nov. 1983.

[3] G. D. Forney, Jr., "Trellis Shaping," *IEEE Trans. Inform. Theory*, Vol. 38, No. 2, March 1992.

[4] G. D. Forney, Jr., "Multidimensional Constellations - Part II: Voronoi Constellations," *IEEE J. Select. Areas Commun.*, Vol. 7, No. 6, August 1989.

[5] K. A. Schouhamer Immink, *Coding Techniques for Digital Recorders*, Prentice Hall (UK).

[6] M. Siala and G. Kawas Kaleh, "RLL Coding Using Structured Vector Quantizing," submitted to IEEE Trans. Inform. Theory, August 1994.

Convergence Behavior and Differential Trellis Code Construction for Decoders Using Multiple Symbol Detection

K. Mackenthun Jr.
2409 Jennifer Ct, Clarksburg, MD 20871 USA

Abstract

A new class of trellis codes is constructed, which aim to perform well with both coherent and noncoherent detection. The codes are constructed by a modification of Ungerboeck's set partitioning approach and use of elementary group theory. Examples of such codes show advantages over M-DPSK trellis codes.

1. Introduction

In this paper, we are concerned with $M = 4$ or $M = 8$, and transmission of 1 bit per symbol duration, or 1 bit/T, for 4PSK, and 2 bit/T for 8PSK. We consider a class $\mathcal{C}$ of trellis codes for 4PSK and 8PSK, consisting of the rate 1/2 binary convolutional codes listed in Table 11.1 of [1], for 4, 8, 16, 32, and 64 states, used with 4PSK modulation by Gray mapping, and the 4, 8, and 16 state trellis codes for 8PSK given by Ungerboeck [2]. By following a code in the class $\mathcal{C}$ with a differential encoder, which is a modulo M adder, we form an M-DPSK trellis code, in a class we refer to as $d(\mathcal{C})$. Suppose the code in class $\mathcal{C}$ has Q states. Then the code in class $d(\mathcal{C})$ may be regarded as having Q superstates, with M phase states in each superstate, for a total of MQ states.

If channel phase is known, we can decode the M-DPSK trellis code with an MQ state decoder, termed *coherent detection.* If channel phase is not known and the receiver uses conventional 2 symbol differential detection, the receiver may still use an MQ state decoder. However, since the metric for conventional differential detection is not sensitive to channel phase, in practice all M phase states in a superstate are lumped together, and a Q state decoder is used, in which only one trellis path is retained for each superstate.

Recently, there has been interest in noncoherent detection which employs more than 2 symbols. In one variation of this approach, branch metrics employ the most recent β channel symbols and β path symbols, termed *multiple symbol detection* (MSD) branch metrics, where $\beta = 2$ for conventional differential detection. For an M-DPSK trellis code, as β increases above 2, we find that performance of an MQ state decoder using MSD branch metrics converges to coherent detection while the performance of the Q state decoder does not converge, and in fact becomes somewhat worse if β increases too much. Besides the reduced number of states, a problem with the Q state decoder is that there is no labelling of decoder states with phases which is consistent with phase changes between the states.

This leads to the question of whether one can directly design trellis codes of small complexity that perform well for both small and large β, and also coherent detection. To address this issue, a general class of *differential trellis codes* $\mathcal{D}$ is constructed. The codes are based on the idea of assigning a signal point to a trellis state, rather than a trellis branch, as done in designing codes for coherent detection. These codes are constructed to have a split and merge property for both 2 symbol detection and coherent detection.

2. Construction of Differential Trellis Codes

In the remainder of the paper, we work with the group $(Z_M)^p$, whose elements are vectors of p components from Z_M, the integers modulo M, or $\{0, 1, \ldots M-1\}$. Group addition is componentwise addition modulo M. An element of $(Z_M)^p$ can be thought of as a p component vector of MPSK symbols, i.e., a $2p$ dimensional signal point. We construct differential trellis codes for $p \geq 1$.

For any sets F, G, and H contained in $(Z_M)^p$, we say $F + G = H$ if $f + G = H$ for all $f \in F$, or in other words $\{f + g | g \in G\} = H$ for all $f \in F$, where '+' denotes group addition. The following elementary result is useful.

Theorem 1. Fix $f \in F$. If $F + G = H$, the mapping $e : G \rightarrow H$ defined by $e(g) = f + g$, is a one to one and onto function. Thus e defines a one to one correspondence between elements of G and H.

Proof. We first show e is a function. Suppose $g_1, g_2 \in G$ with $g_1 = g_2$. Then $e(g_1) = f + g_1 = f + g_2 = e(g_2)$. Thus e is a function. Next, suppose $e(g_1) = e(g_2)$. Then $f + g_1 = f + g_2$ which implies $g_1 = g_2$. Thus e is one to one. Next, suppose $h \in H$. Since $F + G = H$, there is a $g \in G$ such that $f + g = h$, or $e(g) = h$. Thus e is onto.

To construct a differential trellis code, we consider the trellis structure shown in Figure 1a. Figure 1a is composed of *superstates* and *connections* between superstates. Each superstate in Figure 1a is labelled with a set $C0$, $C1$, $C2$, or $C3$, which are obtained by the Ungerboeck partition: $B0 = C0 \cup C2$ and $B1 = C1 \cup C3$, and $A0 = B0 \cup B1$, where $A0 \subset (Z_M)^p$; each set element is a state in the labelled superstate. Suppose the number of elements in each set Ci is m. A trellis of the form in Figure 1a will be called an *m point per superstate q superstate differential trellis*, or m pt/ss q superstate trellis. When $m = 1$, we also refer to this as a 1 point per state q state trellis, or 1 pt/s q state trellis.

A connection between two superstates means that any two states in the superstates, one from each superstate, are connected by a branch. We require that the number of branches out of each state be 2^p for 4PSK and 4^p for 8PSK. Thus for the superstate trellis of Figure 1a, $m = 2^p/2$ for 4PSK and $m = 4^p/2$ for 8PSK.

We can redraw any m pt/ss q superstate trellis as a 1 pt/s mq state trellis by drawing states in each superstate as points, and drawing a branch between two points if the two points are in superstates which are connected. For example, Figure 1b shows a 1 pt/s 16 state trellis, obtained by redrawing the trellis of Figure 1a assuming the sets Ci consist of $m = 2$ points, given by $C0 = \{d_0, d_4\}$, $C1 = \{d_1, d_5\}$, $C2 = \{d_2, d_6\}$, and $C3 = \{d_3, d_7\}$. The trellis of Figure 1b could be used for either 4PSK, with $p = 2$, or 8PSK, with $p = 1$.

Motivated by the work of [2], we require that the superstate trellis satisfy the following three rules:

a. Connections which split from the same superstate go to superstates labelled with points from either the set $B0$ or $B1$.

b. Connections which merge into the same superstate come from superstates labelled with points from either the set $B0$ or $B1$.

c. Use sets $B0$ and $B1$ equally.

It can be seen that the example 8 superstate trellis in Figure 1a satisfies **a**, **b**, and **c**. We add an additional requirement not present in [2].

d. $Bn_1 + Bn_2 = B\{(n_1 + n_2) \text{mod} 2\}$ for $n_1, n_2 \in \{0, 1\}$.

Condition **d** appears easily satisfied with all good partitions. The following condition is sufficient.

Theorem 2. $B0$ is a subgroup of $(Z_M)^p$, and there is an $x \in (Z_M)^p$ such that $B1 = B0 + x$ and $B0 = B1 + x$.

Proof. We have $B0 + B0 = B0$ since for any $f \in B0$ and any $h \in B0$, there is a $g \in B0$ such that $f + g = h$, given by $g = (-f) + h$. $B1 + B1 = (B0 + x) + (B0 - x) = B0$. $B0 + B1 = B0 + (B0 + x) = B1$. $B1 + B0 = (B0 + x) + B0 = B1$.

Suppose superstate S' is labelled with points from Bi and connected to superstates labelled with points from Bj, $i, j \in \{0,1\}$. From **d**, $Bi + B\{(i+j)\text{mod}2\} = Bj$. Suppose $b_i \in Bi$ and b_i is a point in S'. Then from Theorem 1, the mapping $e_{ij} : B\{(i+j)\text{mod}2\} \rightarrow Bj$ defined by $e_{ij}(b') = b_i + b'$ defines a one to one correspondence between group elements $b' \in B\{(i+j)\text{mod}2\}$ and elements of Bj. But there is a branch out of b_i to each point in Bj. Thus there is a one to one correspondence between group elements $b' \in B\{(i+j)\text{mod}2\}$ and branches out of b_i.

Let I be the set of all information vectors of length p. There are 2^p elements in I for 4PSK and 4^p for 8PSK. We now develop a one to one correspondence between elements $\mathbf{i} \in I$ and branches out of each state. Define arbitrary one to one and onto mappings f_0 and f_1, $f_0 : I \rightarrow B0$ and $f_1 : I \rightarrow B1$. Let b_i, Bi, and Bj be as before. Then the mapping $\tilde{e}_{ij} : I \rightarrow Bj$ defined by

$$\tilde{e}_{ij}(\mathbf{i}) = e_{ij}(f_{(i+j)\text{mod}2}(\mathbf{i})) \quad (1)$$

$$= b_i + f_{(i+j)\text{mod}2}(\mathbf{i}) \quad (2)$$

defines a one to one correspondence between information vectors $\mathbf{i} \in I$ and branches out of b_i.

A p component vector of transmitted MPSK symbols, $\mathbf{s}$, $\mathbf{s} = (a_{k+1}, a_{k+2}, \ldots a_{k+p})$, can be represented as a point in $(Z_M)^p$. Thus as we travel from state to state in the trellis, we obtain a sequence of transmitted vectors $\mathbf{s}_1, \mathbf{s}_2, \ldots$. The branch we take to get from one state to the next depends on the sequence of information vectors $\mathbf{i}_1, \mathbf{i}_2, \ldots$ and the mapping $\tilde{e}_{ij}$. Assume we have just transmitted $\mathbf{s}_n$, which is assigned to a superstate labelled with points from Bi, and the superstate is connected to superstates labelled with points from Bj. Using (2), for an input information vector $\mathbf{i}_n$, the next transmitted vector $\mathbf{s}_{n+1}$ is obtained as

$$\mathbf{s}_{n+1} = \mathbf{s}_n + f_{(i+j)\text{mod}2}(\mathbf{i}_n). \quad (3)$$

For use with noncoherent detection, we must ensure that codes are *rotationally complete*, that is, if there are two codewords which are a rotation of one another, they both decode to the same information sequence. Rotationally invariant codes [3] are rotationally complete. Here we show that the code obtained from the superstate trellis in the manner just described is rotationally invariant. The code is rotationally invariant if the superstate trellis has the following property: branches connecting any two signal point pairs which are rotations of one another are assigned the same information vector. More precisely, if $\mathbf{i}_n$ is assigned to the branch connecting $\mathbf{s}_n$ and $\mathbf{s}_{n+1}$, and $\mathbf{i}'_n$ is assigned to the branch connecting $\mathbf{s}'_n$ and $\mathbf{s}'_{n+1}$, where $\mathbf{s}'_n = \mathbf{s}_n + y$ and $\mathbf{s}'_{n+1} = \mathbf{s}_{n+1} + y$, $y \in (Z_M)^p$, then we must have $\mathbf{i}_n = \mathbf{i}'_n$. We now show the above method of assigning information vectors satisfies this property. Assume $\mathbf{s}_n \in Bi$, $\mathbf{s}_{n+1} \in Bj$, $\mathbf{s}'_n \in Bi'$, and $\mathbf{s}'_{n+1} \in Bj'$. From (3), we have $f_{(i+j)\text{mod}2}(\mathbf{i}_n) = \mathbf{s}_{n+1} - \mathbf{s}_n$ and $f_{(i'+j')\text{mod}2}(\mathbf{i}'_n) = \mathbf{s}'_{n+1} - \mathbf{s}'_n = \mathbf{s}_{n+1} - \mathbf{s}_n$. From **d**, we have $\mathbf{s}_{n+1} - \mathbf{s}_n \in B0$ or $\mathbf{s}_{n+1} - \mathbf{s}_n \in B1$. If $\mathbf{s}_{n+1} - \mathbf{s}_n \in B0$, then $(i+j)\text{mod}2 = (i'+j')\text{mod}2 = 0$, and since f_0 is one to one, we have $\mathbf{i}_n = \mathbf{i}'_n$. By similar reasoning, if $\mathbf{s}_{n+1} - \mathbf{s}_n \in B1$ we also have $\mathbf{i}_n = \mathbf{i}'_n$, and thus we have shown the codes are rotationally invariant.

The last concern is that the code be noncatastrophic. A code is catastrophic if it produces two infinite length codeword sequences which differ in only a finite number of symbols but which decode to two information sequences which differ in an infinite number of positions [1]. But the differential trellis code cannot be catastrophic because

where two codeword sequences agree they have the same sequence of signal points and by the construction for rotational invariance decode to the same sequence of information vectors.

Codes which satisfy **a**, **b**, **c**, and **d**, and which assign information vectors to branches in the manner just discussed, form the class $\mathcal{D}$ of differential trellis codes.

So far, we have constructed a q superstate differential trellis code of mq states which is rotationally invariant and non catastrophic. Such a code may be decoded with an mq state decoder using coherent detection or MSD branch metrics. We now derive a novel MSD branch metric for use with mq state decoding. Let $\hat{\mathbf{a}}_k(i), i = 1, \ldots mq$ be the survivor into state i at trellis stage k, where $\hat{\mathbf{a}}_k(i) = (\hat{a}_0(i), \ldots \hat{a}_k(i))$ are the 0^{th} through k^{th} symbols in the survivor path. To find the mq survivors at the next trellis stage $k+p$ (for notational convenience, we number the trellis stages $\ldots k, k+p, k+2p, \ldots$), we extend the mq survivors at stage k to obtain tentative survivors of the form $\tilde{\mathbf{a}}_{k+p} = (\hat{\mathbf{a}}_k(i), \tilde{a}_{k+1}, \ldots \tilde{a}_{k+p})$, where $\tilde{\mathbf{a}}_{k+p}$ is an extension of $\hat{\mathbf{a}}_k(i)$ with vector $(\tilde{a}_{k+1}, \ldots \tilde{a}_{k+p})$. Among tentative survivors which merge in the same state at stage $k+p$, we select the winning survivor by finding $(\tilde{a}_{k+1}, \ldots \tilde{a}_{k+p})$ which maximizes $p(\mathbf{r}_{k+p}|\tilde{\mathbf{a}}_{k+p})$, where $\mathbf{r}_{k+p} = (r_0, \ldots r_{k+p})$ are the received samples up to time $k+p$, over an AWGN channel with unknown phase. By manipulating $p(\mathbf{r}_{k+p}|\tilde{\mathbf{a}}_{k+p})$, truncating to β channel symbols, and then averaging over channel phase, we obtain a branch metric for p component vectors:

$$\ln p(r_{k+1}, \ldots r_{k+p}|\mathbf{r}_k^{\beta-p}, \tilde{\mathbf{a}}_{k+p}^{\beta}) \tag{4}$$

$$\sim \ln I_o\left(\frac{1}{\sigma^2}\left|S(\hat{\mathbf{a}}_k^{\beta-p}(i)) + r_{k+1}\tilde{a}_{k+1}^* + \ldots r_{k+p}\tilde{a}_{k+p}^*\right|\right) - \ln I_o\left(\frac{1}{\sigma^2}\left|S(\hat{\mathbf{a}}_k^{\beta-p}(i))\right|\right), \tag{5}$$

where for any vector $\mathbf{x} = (x_0, \ldots x_k, \ldots)$, and any integers k, η, $\mathbf{x}_k^{\eta}$ is defined by $\mathbf{x}_k^{\eta} = (x_{k-\eta+1}, \ldots x_k)$ and $S(\mathbf{x}_k^{\eta})$ is defined by $S(\mathbf{x}_k^{\eta}) = \sum_{n=k-\eta+1}^{k} r_n x_n^*$. We assume p divides β evenly. Branch metric (4) is difficult to implement, and simulation results show there is no loss in using the following high SNR approximation:

$$\ln p(r_{k+1}, \ldots r_{k+p}|\mathbf{r}_k^{\beta-p}, \tilde{\mathbf{a}}_{k+p}^{\beta}) \sim |S(\hat{\mathbf{a}}_k^{\beta-p}(i)) + r_{k+1}\tilde{a}_{k+1}^* + \ldots r_{k+p}\tilde{a}_{k+p}^*| - |S(\hat{\mathbf{a}}_k^{\beta-p}(i))|. \tag{6}$$

We may also decode a q superstate differential trellis code with a q state decoder using (6). To ensure that q state decoders work well, we add a restriction on the sets Ci, which appears easily satisfied for good partitions:

e. For $n_1, n_2 \in \{0,1,2,3\}$, $Cn_1 + Cn_2 = Cj$ for some j, $j \in \{0,1,2,3\}$.

Theorem 3. Assume **d** and **e** are satisfied. Then for any pair n_1, j, with $n_1, j \in \{0,1,2,3\}$, the equation $Cn_1 + Cx = Cj$, viewed as an unknown in x, has one and only one solution.

Proof. First, we show that $C0 + Cx = Cj$ has one and only one solution for any $j \in \{0,1,2,3\}$. By **e**, we have $C0 + C0 = C\gamma$ and $C0 + C2 = C\delta$, for some γ, δ, with $\gamma, \delta \in \{0,1,2,3\}$. Since by **d**, $B0 + B0 = B0$, we have $\gamma \in \{0,2\}$ and $\delta \in \{0,2\}$. We cannot have $\gamma = \delta$ since this would imply $C0 + B0 = C\gamma$, which contradicts $B0 + B0 = B0$. Thus $\gamma \neq \delta$. Next, by **e**, we have $C0 + C1 = C\zeta$ and $C0 + C3 = C\xi$. Since by **d**, $B0 + B1 = B1$, we have $\zeta \in \{1,3\}$ and $\xi \in \{1,3\}$. We cannot have $\zeta = \xi$ since this would imply $C0 + B1 = C\zeta$, which contradicts $B0 + B1 = B0$. Thus $\zeta \neq \xi$. This establishes that the equation $C0 + Cx = Cj$ has one and only one solution for any $j \in \{0,1,2,3\}$. Similarly, we can show that for $n = 1,2,3$, the equation $Cn + Cx = Cj$ has one and only one solution for any $j \in \{0,1,2,3\}$.

For q state decoding, an important implication of **e** and Theorem 3 is the following. Suppose superstate S' is connected to superstate S''', and S' is labelled with set Cv and S''' is labelled with set Cw. By **e** and Theorem 3, there is a unique set Cu which satisfies $Cv + Cu = Cw$. Then applying Theorem 1, this means that for any point $c_v \in Cv$, there is a one to one correspondence between group elements $\mathbf{u} \in Cu$ and branches out of c_v which go to states in S'''. To indicate this relation, we label the trellis connection between Cv and Cw with Cu. Figure 2a and 2b show the trellis of Figure 1a labelled in this manner, with 2 possible labellings of trellis connections which satisfy **e** and Theorem 3.

The fact that each connection of a q superstate trellis can be labelled with a unique set Ci makes effective q state decoding possible, as follows. In q state decoding, a single winning trellis path is kept for each superstate. Assume the winning trellis path at stage k is $\hat{\mathbf{a}}_k(i)$ for superstate S_i, $i = 1, \ldots q$. We now discuss how to find the winning trellis paths at stage $k+p$. Suppose connections from superstates S' and S'' merge into superstate S'''. Assume the winning trellis path into S' at stage k is $\hat{\mathbf{a}}_k(i')$, and the connection from S' to S''' is labelled with set C'. Assume the winning trellis path into S'' at stage k is $\hat{\mathbf{a}}_k(i'')$, and the connection from S'' to S''' is labelled with set C''. The winning path for superstate S''' at stage $k+p$ is the path which maximizes the metric (6), chosen from among paths in the set $\{(\hat{\mathbf{a}}_k(i), \tilde{a}_{k+1}, \ldots \tilde{a}_{k+p})\}$, where i takes values in $\{i', i''\}$, and for $i = i'$, the tentative vector $(\tilde{a}_{k+1}, \ldots \tilde{a}_{k+p})$ takes all possible values in the set $\{(\hat{a}_{k-p+1}(i')e^{j2\pi u_1/M}, \ldots \hat{a}_k(i')e^{j2\pi u_p/M}) | \mathbf{u} \in C'\}$, and for $i = i''$, the tentative vector $(\tilde{a}_{k+1}, \ldots \tilde{a}_{k+p})$ takes all possible values in the set $\{(\hat{a}_{k-p+1}(i'')e^{j2\pi u_1/M}, \ldots \hat{a}_k(i'')e^{j2\pi u_p/M}) | \mathbf{u} \in C''\}$, where $\mathbf{u} = (u_1, \ldots u_p)$.

Of course, Figure 1a is not the only example of a superstate trellis which satisfies **a**, **b**, and **c**. Figure 3a shows a 32 superstate trellis which satisfies **a**, **b**, and **c**, with connections out of one superstate going to 4 other superstates, where partition sets Dl, $l = 0, \ldots 7$, satisfy $Cj = Dj \cup D\{j+4\}$, $j = 0, 1, 2, 3$. Clearly, we can also construct codes with connections out of one superstate going to 8 other superstates, and so on.

Further, we can design trellises with a different number of superstates than for the two examples in Figure 1a and 3a. For connections from one superstate to 2 superstates, codes can be found which meet rules **a**, **b**, and **c** for $8, 16, 32, \ldots$ superstates. Figure 3b shows an example for 16 superstates. For 4 superstates, we cannot satisfy **a**, **b**, and **c**, but instead **c** and either **a** or **b**. With connections from one superstate going to 4 other superstates, rules **a**, **b**, and **c** can be satisfied for $16, 32, \ldots$ superstates. Figure 1b shows an example for 16 superstates, obtained by regarding each point d_j in Figure 1b as a superstate labelled with set D_j.

Ungerboeck has constructed codes for coherent detection using a trellis of states and branches [2]. We say a code is an *Ungerboeck code* if it satisfies the following three rules [2]:

a1. Branches which split from the same state are labelled with points from either the set $B0$ or $B1$.

b1. Branches which merge into the same state are labelled with points from either the set $B0$ or $B1$.

c1. Use sets $B0$ and $B1$ equally.

Codes in the class $\mathcal{C}$ are Ungerboeck codes, where we use $B0 = \{0, 2\}$ and $B1 = \{1, 3\}$ for 4PSK and $B0 = \{0, 2, 4, 6\}$ and $B1 = \{1, 3, 5, 7\}$ for 8PSK. As seen in Section 1, a differential encoding of a Q state code in $\mathcal{C}$ results in an M pt/ss Q superstate code in class $d(\mathcal{C})$. The M pt/ss Q superstate trellis can be redrawn as an MQ state trellis in

which each state is assigned one phase. Since the trellis of the code in class $\mathcal{C}$ satisfies **a1**, **b1**, and **c1**, the MQ state trellis of the corresponding code in class $d(\mathcal{C})$ can be regarded as an MQ superstate trellis which satisfies **a**, **b**, and **c**. As an example, Figure 3c shows the 4 pt/ss 4 superstate trellis obtained from the differential encoding of the rate 1/2 4 state 4PSK code in class $\mathcal{C}$. Each point in a superstate is a phase state of phase v, $v \in \{0,1,2,3\}$. By regarding each phase state v as a superstate labelled with set Cv, we obtain a 16 superstate trellis which satisfies **a**, **b**, and **c**.

Since Figure 3b and 3c both represent 16 superstate trellises, it is natural to ask if they are the same. However, note that the 16 superstate trellis resulting from Figure 3c cannot be rearranged into the 16 superstate trellis in Figure 3b, since there are 4 states in Figure 3c which branch to themselves, whereas there are only 2 superstates in Figure 3b that connect to themselves. Lastly, note that there is no rate 1/2 2 state 4PSK code which satisfies **a1**, **b1**, and **c1**, and thus there is no 8 superstate trellis which is a differential encoding of a 2 state trellis and which satisfies **a**, **b**, and **c**.

Although with appropriate interpretation the MQ state trellis of codes in $d(\mathcal{C})$ satisfies **a**, **b**, and **c**, codes in $d(\mathcal{C})$ are not in $\mathcal{D}$ because information vectors are assigned to branches of $\mathcal{C}$, which results in a different assignment to branches of $d(\mathcal{C})$ than for codes in $\mathcal{D}$. However, codes in $d(\mathcal{C})$ can be easily modified to belong to $\mathcal{D}$ by assigning information vectors to branches of the MQ state trellis, in the same manner as done in construction of codes in $\mathcal{D}$. In fact, since an error event for 2 symbol detection is likely to have many rotations between successive signal points which are the same as the correct path, it is perhaps better that M-DPSK trellis codes are modified so that these rotations are assigned the same information bit(s), as done for codes in $\mathcal{D}$.

In the next section, we give some examples of differential trellis codes.

3. Examples

A. Two Dimensional 1 pt/ss Codes for 4PSK and 8PSK

We can construct a 2D 1 pt/ss 8 superstate differential trellis code for 4PSK using the superstate trellis of Figure 1a, with the partition: $C0 = \{0\}$, $C2 = \{2\}$, $C1 = \{1\}$, and $C3 = \{3\}$, with $B0 = C0 \cup C2$ and $B1 = C1 \cup C3 = B0 + 1$. It is easy to show that rules **d** and **e** are met for this partition, with connection labellings as shown in Figure 2a. The code has 8 states with one phase per state.

Using Figure 2a and the indicated partition, it is seen that any state splits to two phases which are antipodal, and the merges into a state come from phases which are antipodal. Thus, the code has an optimum split and merge property for coherent detection.

Further, it is seen that the phase changes out of any state are either from the set $B0 = \{0,2\}$, or the set $B1 = \{1,3\}$, and the merges into any state are from either $B0$ or $B1$. From consideration of the metric for conventional two symbol differential detection, or metric (6), it is seen that these are the optimum phase changes for a split and merge, and therefore the code has an optimum split and merge property for two symbol detection. Therefore the differential trellis code should work well for both coherent detection and two symbol detection. Simulation results confirm that compared to other 8 state decoders, the 8 state differential trellis code performs well for both 2 symbol detection, large β MSD, and coherent detection. Compared to 8 state decoding of the 32 state 4-DPSK code in $d(\mathcal{C})$, the differential trellis code gives comparable performance for $\beta = 2$ and greatly improved performance for large β.

Let d^2_{free} be the minimum squared Euclidean distance between the correct trellis path and any error event path which splits from the correct path. (Symbols are points on the

unit circle.) Then d^2_{free} of the 1 pt/s 8 state differential trellis code is 10, while d^2_{free} of the 8 state 4PSK code of [1] is 12. Thus the asymptotic coherent detection performance of the code of [1] is .79 dB better. (Note that d^2_{free} of an 8 state 4-DPSK code is 6.)

It is easy to understand why coherent detection of the 8 state differential trellis code does not perform quite as well as coherent detection of the 8 state code of [1]. Consider an error event which splits from the correct path for l branches and then remerges. For the 8 state code of [1], phases are assigned to branches, of which there are l unmerged branches, while for the differential trellis code, phases are assigned to states, of which there are $l-1$ unmerged states. Therefore, all other things being equal, differential trellis codes should not perform quite as well with coherent detection as codes in $\mathcal{C}$. By the same token, however, it might be expected that as the number of states increases so that d^2_{free} and thus l become large, the two types of codes should perform about the same.

By using the same partition: $C0 = \{0\}$, $C2 = \{2\}$, $C1 = \{1\}$, and $C3 = \{3\}$, with the 16 superstate trellis of Figure 2b, we obtain a 1 pt/s 16 state 4PSK differential trellis code with $d^2_{free} = 12$. By comparison, the 16 state 4-DPSK trellis code of Figure 3c has $d^2_{free} = 10$. It is easy to construct similar 4PSK differential trellis codes for more than 16 states.

We can construct a 2D 1 pt/s 32 state differential trellis code for 8PSK using the 32 superstate trellis of Figure 3a, with the partition: $Di = \{i\}, i = 0, \ldots 7$. It is easy to show that rule **d** is met. We use a modification of **e**, which applies to this code for the sets Di, as follows: For $n_1, n_2 \in \{0, 1, \ldots 7\}$, $Dn_1 + Dn_2 = Dj$ for some $j, j \in \{0, 1, \ldots 7\}$. Similar codes can be constructed for more than 32 states. In the next section, we construct a 16 state code for 8PSK, based on an 8 superstate trellis.

B. Two Dimensional 2 pt/ss Codes for 8PSK

We can construct a 2D 2 pt/ss 8 superstate differential trellis code for 8PSK using the superstate trellis of Figure 1a, with the partition: $C0 = \{0,4\}$, $C2 = \{2,6\}$, $C1 = \{1,5\}$, and $C3 = \{3,7\}$, with $B0 = C0 \cup C2$ and $B1 = C1 \cup C3 = B0 + 1$. It is easy to show that rules **d** and **e** are met for this partition, with the connection labelling as shown in Figure 2a. The equivalent 1 pt/s 16 state trellis is shown in Figure 1b, after using the correspondence $d_i \leftrightarrow i$, $i = 0, \ldots 7$. Using Figure 1b, we find that d^2_{free} is limited to 4 for this code.

Simulation results for $q = 8$ state decoding performance using the high SNR MSD branch metric show that performance approaches coherent detection performance for large β, even though a 16 state decoder is used for coherent detection. Compared to performance of the 64 state 8-DPSK trellis code in class $d(\mathcal{C})$, for $\beta = 2$ performance of the differential trellis code is perhaps 0.1 dB better, while for large β performance is much better. In fact, 8 state decoder performance of the differential trellis code for $\beta = 16$ is only 0.6 dB worse than 64 state coherent detection of the 8-DPSK trellis code.

It is easy to construct similar codes for more than 8 superstates.

C. Multi-Dimensional m pt/ss Codes

Using the 8 superstate trellis of Figure 1a we may construct multidimensional differential trellis codes for 4PSK and 8PSK. To construct a 4D code for 4PSK, we use the set partition: $C0 = \{00,22\}$, $C2 = \{02,20\}$, $C1 = \{11,33\}$, and $C3 = \{13,31\}$, with $B0 = C0 \cup C2$ and $B1 = C1 \cup C3 = B0 + 11$. It is easy to show that rules **d** and **e** are met for this partition, with the connection labelling as shown in Figure 2b. For this code, we have $d^2_{free} = 8$.

To construct an 8D code for 4PSK, we use the partition:

$$\begin{aligned} C0 &= \{0000, 0202, 0022, 0220, 2222, 2020, 2200, 2002\}, \\ C2 &= \{1111, 1313, 1133, 1331, 3333, 3131, 3311, 3113\}, \\ C1 &= \{0110, 0312, 0132, 0330, 2332, 2130, 2310, 2112\}, \\ C3 &= \{1221, 1023, 1203, 1001, 3003, 3201, 3021, 3223\}, \end{aligned}$$

with $B0 = C0 \cup C2$ and $B1 = C1 \cup C3 = B0 + 0110$. The set $B0$ is formed from a binary Reed-Muller code of 8 bits, by Gray mapping pairs of bits into 4PSK symbols; by work of [4], the set $B0$ forms a subgroup of $(Z_4)^4$. It is easy to show that rules **d** and **e** are met for this partition, with the connection labelling as shown in Figure 2b. We have $d^2_{free} = 8$. Because d^2_{free} of the 8 superstate multidimensional codes for 4PSK is limited to 8, while d^2_{free} of the 2D 8 state 4PSK differential trellis code of Section A is 10, the 4D and 8D multidimensional codes do not perform quite as well as the 2D code. However, the multidimensional codes may have some advantages for use on channels with ISI. Furthermore, note that the $q = 8$ state decoding complexity of the multidimensional codes may be somewhat less than the 2D code, since for example we decode 8 states per 8D vector for the 8D code, while we decode 8 states per 2D point for the 2D code. Lastly, note that by further subdividing the 8D partition, we may construct an 8D 2pt/ss 128 superstate code with 8 connections out of each superstate, having $d^2_{free} = 16$.

To construct a 4D 8 pt/ss 8 superstate differential trellis code for 8PSK, we use the trellis of Figure 1a, and a 4D partition shown in Figure 3 of [3], given by:

$$\begin{aligned} C0 &= \{00, 22, 44, 66, 04, 26, 40, 62\}, \\ C2 &= \{24, 46, 60, 02, 20, 42, 64, 06\}, \\ C1 &= \{11, 33, 55, 77, 15, 37, 51, 73\}, \\ C3 &= \{13, 35, 57, 71, 17, 31, 53, 75\}, \end{aligned}$$

with $B0 = C0 \cup C2$ and $B1 = C1 \cup C3 = B0 + 11$. It is easy to show that rules **d** and **e** are met for this partition, and that the connection labelling is as shown in Figure 2b. We have $d^2_{free} = 4$ for this code, which is the same d^2_{free} as for the 2D 8PSK code. Compared to performance of the 2D 2 pt/ss 8 superstate code, the performance of the 4D code is several tenths of a dB worse for the same type of decoder setup.

Using the principles illustrated so far, other multidimensional codes can be constructed.

References

[1] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1983, p. 330.

[2] G. Ungerboeck, "Channel Coding with Multilevel/Phase Signals," *IEEE Transactions on Information Theory*, Vol. IT-28, No. 1, January 1982, pp. 55-67.

[3] L.-F. Wei, "Rotationally Invariant Trellis-Coded Modulations with Multidimensional M-PSK," *IEEE Journal on Selected Areas in Communications*, Vol. 7, December 1989, pp. 1281-1295.

[4] A. R. Hammons, Jr., et. al., "The Z_4-Linearity of Kerdock, Preparata, Goethals, and Related Codes," *IEEE Transactions on Information Theory*, Vol. 40, No. 2, March 1994, pp. 301-319.

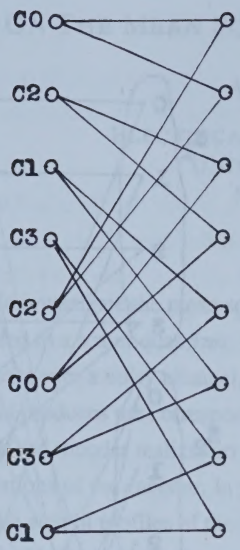


Fig (1a) 8 superstate differential trellis

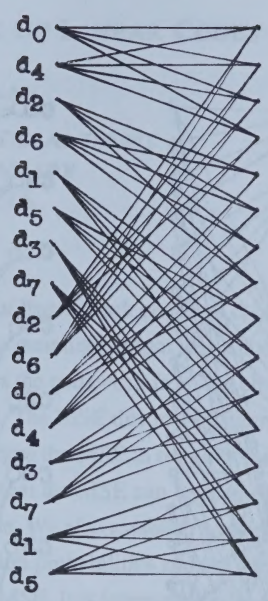


Fig (1b) Trellis of (1a) redrawn for $m = 2$

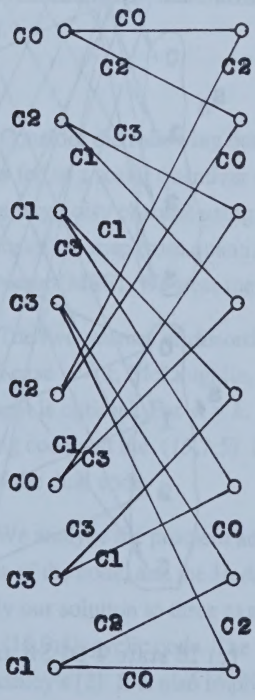


Fig (2a) One connection labelling

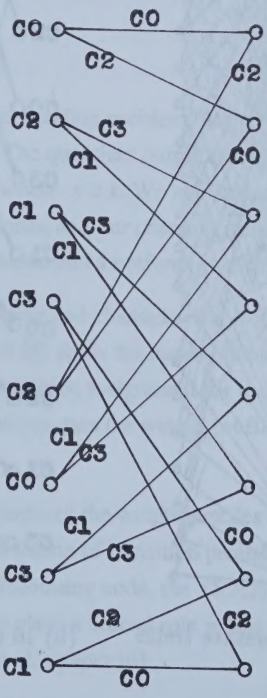
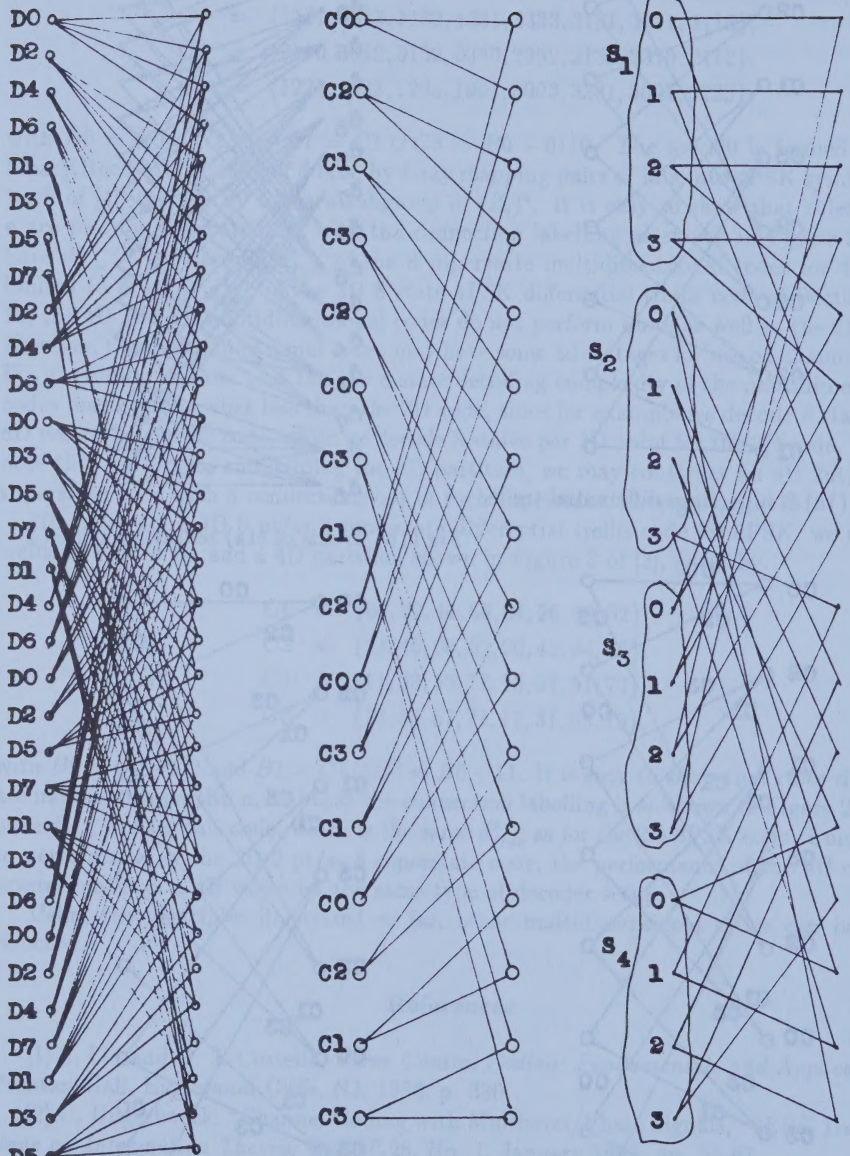


Fig (2b) Another labelling



(a) 32 superstate trellis (b) 16 superstate trellis (c) 16 state 4-DPSK trellis code

Fig (3) Examples of superstate trellis

ON THE MEAN SQUARED ERROR OF LINEAR CODES

ALI S. KHAYRALLAH
ELECTRICAL ENGINEERING DEPARTMENT
UNIVERSITY OF DELAWARE
NEWARK, DE 19716

ABSTRACT

We investigate the mean squared error (MSE) of linear block codes. A continuous source is fed to a scalar quantizer. An encoder maps the quantizer output into a codeword, which is sent over a noisy channel. A decoder estimates the codeword from the channel output, and produces the corresponding quantizer level. The MSE due to the channel is a function of the encoder realization of the code. We analyze the MSE in terms of the input / output relations of the encoder. In particular, we show that the MSE can be expressed in terms of the weight profiles of the subspaces of the code. We also establish a relation between the Hadamard transform of the channel transition probabilities and the MSE. We use our analysis to find the encoder that minimizes the MSE. It is interesting to note that the encoder realization that minimizes the MSE can change with the raw error rate of the channel.

INTRODUCTION

Consider the following basic joint source channel coding problem. A continuous source is fed to a scalar quantizer with $K = 2^k$ levels. The quantizer output is transmitted over a binary noisy channel using an n-bit codeword, where $n \geq k$. We are concerned with the choice of the map from quantizer outputs to codewords, and our criterion is the mean squared error (MSE). We treat the case of a uniform source and a uniform quantizer.

The two sources of distortion are quantization error and channel error. For the uncoded case $n = k$, McLaughlin, Ashley, and Neuhoff [6] show the natural binary assignment is optimal. For $n > k$, Ashley [2] finds the optimal assignment for the (7,4;3) Hamming code and the (15,7;5) BCH code. His technique uses the weight profiles of the cosets of the dual code.

We analyze the problem and find a solution in terms of the weight profiles of the subcodes of the code, and the Hadamard transform of a matrix of transition probabilities. We apply our solution to three examples, the (7,4;3) Hamming code, the (5,3;2) code, and the (15,9;4) cyclic code. The Hadamard transform plays a crucial role in this work and in Ashley's [2]. It is also implicit in McLaughlin et al.'s paper [6].

BACKGROUND

The source is uniform over $[-1/2,+1/2]$. Its output is fed to a uniform quantizer with K levels, which outputs a level q. The index assignment B is a one-to-one map from q to $\mathbf{x} \in \{0,1\}^k$.

The $(n,k;d)$ binary linear code has input length k, output length n, and minimum distance d. The relevant distance is the familiar Hamming distance $d(\mathbf{u},\mathbf{v})$. We also use the Hamming weight $w(\mathbf{u})$. The encoder is described by a $(k \times n)$ generator matrix $\mathbf{G}$, which maps $\mathbf{x}$ into a codeword $\mathbf{y} = \mathbf{x}\,\mathbf{G}$. The codebook C is defined as the set of codewords. The channel is the familiar binary symmetric channel with raw error rate ε. At the output of the channel, the decoder can be described in two stages. The first stage is a decision rule $\mathbf{D}$. Here, $\mathbf{D}$ is a bounded distance decoder [5] with radius $\rho = \lfloor (d-1)/2 \rfloor$. If $\mathbf{z}$ falls within ρ of a codeword $\hat{\mathbf{y}}$, then $\mathbf{D}(\mathbf{z}) = \hat{\mathbf{y}}$. Otherwise, $\mathbf{z}$ falls in the detection region R, and $\mathbf{D}$ outputs an erasure. The second stage of the decoder is an "uncoder". A codeword $\hat{\mathbf{y}} \in C$ is mapped into its corresponding input $\hat{\mathbf{x}}$, and an erasure is mapped into itself. The last stage maps $\hat{\mathbf{x}}$ into its corresponding quantizer level, and an erasure into zero.

Let $\Gamma(C)$ be the set of all $\mathbf{G}$ that produce a codebook C. Distinct matrices in $\Gamma(C)$ differ in their input / output relations. It is easy to see that for any $\mathbf{G}$ and $\mathbf{G}'$ in $\Gamma(C)$, there is a unique $(k \times k)$ matrix $\mathbf{F}$ such that $\mathbf{G}' = \mathbf{F}\,\mathbf{G}$. We refer to $\mathbf{F}$ as a precoder matrix.

Recall that the probability of decoding $\hat{\mathbf{y}}$ given that $\mathbf{y}$ is transmitted depends only on their distance u, and we write $P(\hat{\mathbf{y}} \mid \mathbf{y}) = \pi_u$. An expression for π_u is given in [3]. Let a_j denote the number of codewords of weight j. We refer to a_j, $j = 0,\ldots,n$, as the weight profile of C. The probability of correct decoding is given by $P_c = \pi_0$. The probability of undetected error is given by $P_u = \sum_{i=1}^{n} a_i \pi_i$. We also let $P_{c,u} = P_c + P_u$. The probability of detected error, or erasure, is given by $P_d = 1 - P_{c,u}$.

For a uniform source and a uniform quantizer, the total MSE E_t can be written as

$$E_t = E_s + E_c$$

where $E_s = K^{-2} / 12$ is the MSE due to the quantizer, and E_c is the MSE due to the channel. Furthermore, we write E_c as

$$E_c = E_d + E_u$$

where E_d is due to erasures, and E_u is due to undetected errors. We define the $(K \times K)$ transition matrix $\mathbf{P}$ as

$$P_{i,j} = \pi_{w(\mathbf{y}_i+\mathbf{y}_j)}$$

We also write the quantizer outputs as a column vector $\mathbf{q}$. It can be shown that

$$E_u = \frac{1}{6} P_{c,u} (1 - K^{-2}) - 2 K^{-1} \mathbf{q}^T \mathbf{P} \mathbf{q}$$

and

$$E_d = \frac{1}{12} P_d (1 - K^{-2})$$

Note that E_s is not a function of the code, while E_d is a function of the code but not of the specific realization $\mathbf{G}$. Only the quadratic form $\mathbf{q}^T \mathbf{P} \mathbf{q}$ in E_u is a function of $\mathbf{G}$. We wish to find the minimum E_u^{min} of E_u over $\Gamma(C)$, and the matrix $\mathbf{G}^{min}$ that achieves it.

SUBSPACES, EIGENVALUES, AND THE HADAMARD TRANSFORM

We define the inner product of two row vectors of equal length as

$$\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{u} \mathbf{v}^T \bmod 2$$

For convenience, we let the k-tuple $\mathbf{x}_i$ be the natural binary expansion of integer i. We also let $\mathbf{y}_i = \mathbf{x}_i \mathbf{G}$. We define the subsets

$$C_i = \{\mathbf{y}_j : \langle \mathbf{x}_j, \mathbf{x}_i \rangle = 0\}$$

Let $a_{i,j}$ denote the weight profile of C_i, and let

$$\theta_i = \sum_{\mathbf{y}_j \in C_i} \pi_{w(\mathbf{y}_j)} = \sum_{j=0}^{n} a_{i,j} \pi_j \qquad i = 0,\ldots,2^k-1$$

We classify the C_i's according to their weight profile. Let M be the number of distinct profiles, and let $\bar{a}_{i,j}$ denote the i-th distinct profile, $i = 0,\ldots,M-1$. Also let μ_i be the number of subspaces with the i-th distinct profile. Finally let $\bar{\theta}_i = \sum_{j=0}^{n} \bar{a}_{i,j} \pi_j$, $i = 0,\ldots,M-1$. We assume that the $\bar{\theta}_0 \geq \bar{\theta}_1 \geq \ldots \geq \bar{\theta}_{M-1}$. It can be checked that $\bar{a}_{0,j} = a_j$ for all j. Hence $\bar{\theta}_0 = \theta_0$, and $\mu_0 = 1$.

EXAMPLE 1

The (7,4;3) Hamming code has generator polynomial $g(D) = 1 + D + D^3$. The corresponding generator matrix is

$$\mathbf{G}^{cyc} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

In Table 1, we list the distinct subspace weight profiles, along with μ_i and $\bar{\theta}_i$ (and $\bar{\lambda}_i$ to be explained next) evaluated at $\varepsilon = 0.1$.

THE HADAMARD TRANSFORM

The $(K \times K)$ Hadamard matrix $\mathbf{H}$ is defined as $H_{i,j} = (-1)^{<\mathbf{x}_i, \mathbf{x}_j>}$. Its inverse is given by $\mathbf{H}^{-1} = K^{-1}\mathbf{H}$. The transition matrix $\mathbf{P}$ defined above is binary circulant, by which we mean that

$$P_{i,j} = \pi_{w((\mathbf{x}_i+\mathbf{x}_j)\mathbf{G})} = P_{0,i\oplus j}$$

where $l = i \oplus j$ denotes the integer such that $\mathbf{x}_l = \mathbf{x}_i + \mathbf{x}_j$. It can be shown [1] that $\mathbf{H}$ diagonalizes $\mathbf{P}$, that is

$$\mathbf{H}\,\mathbf{P} = \Lambda\,\mathbf{H}$$

where $\Lambda = \text{diag}\,(\lambda_0, \lambda_1, \ldots, \lambda_{2^k-1})$ and λ_i is the i-th eigenvalue of $\mathbf{P}$. Let $\mathbf{p}$ denote the 0-th column of $\mathbf{P}$, and $\mathbf{h}_i$ denote the i-th row of $\mathbf{H}$. It follows that

$$\lambda_i = \mathbf{h}_i\,\mathbf{p}$$

It can be checked that $\lambda_i = 2\,\theta_i - \theta_0$. Thus except in degenerate cases, there are M distinct eigenvalues

$$\bar{\lambda}_i = 2\,\bar{\theta}_i - \bar{\theta}_0 \qquad\qquad i = 0,\ldots,M-1$$

Also, the multiplicity of $\bar{\lambda}_i$ is equal to μ_i, and the $\bar{\lambda}_i$'s are ranked in decreasing order.

It can also be checked that the Hadamard transform $\tilde{\mathbf{q}} = \mathbf{H}\,\mathbf{q}$ of the uniform quantizer vector $\mathbf{q}$ is given by

$$\tilde{q}_i = \begin{cases} -2^{s-1} & i = 2^s \\ 0 & \text{else} \end{cases}$$

Hence we can write

$$E_u = \frac{1}{6}\,(1 - K^{-2})\,(\lambda_0 - \ell)$$

where

$$\ell = \frac{3}{K^2 - 1} \sum_{s=0}^{k-1} 4^s\,\lambda_{2^s} = \frac{6}{K^2 - 1} \sum_{s=0}^{k-1} 4^s\,\theta_{2^s} - \theta_0$$

Since $\sum_{s=0}^{k-1} 4^s = (K^2 - 1)/3$, ℓ is a weighted average of λ_{2^s}, $s = 0,\ldots,k-1$. Also, E_u is bounded from below by

$$E_u^{lo} = \frac{1}{6}(1-K^{-2})(\bar{\lambda}_0 - \bar{\lambda}_1)$$

and from above by

$$E_u^{up} = \frac{1}{6}(1-K^{-2})(\bar{\lambda}_0 - \bar{\lambda}_{M-1})$$

Note that for $i = 2^{k-1-s}$, $s = 0,...,k-1$, $\langle \mathbf{x}_j, \mathbf{x}_{2^{k-1-s}} \rangle = x_{j,s}$, the bit at position s in $\mathbf{x}_j$. Hence

$$C_{2^{k-1-s}} = \{\mathbf{y}_j : x_{j,s} = 0\}$$

In other words, $C_{2^{k-1-s}}$ is generated by the matrix consisting of $\mathbf{G}$ with row s removed. Let x_s and $\hat{x}_s$ denote the bit at position s in $\mathbf{x}$ and $\hat{\mathbf{x}}$, respectively. Note that if the decoded codeword $\hat{\mathbf{y}}$ belongs to $C_{2^{k-1-s}}$, then $\hat{x}_s = x_s$. Hence $\theta_{2^{k-1-s}}$ is the probability of correctly receiving x_s. In the expression for E_u (via ℓ), x_s is given weight 4^s. This reflects the higher sensitivity of MSE to most significant bits than to least significant bits. Thus the generator matrix that minimizes E_u must satisfy $\lambda_{2^{k-1}} \geq \lambda_{2^{k-2}} \geq ... \geq \lambda_1$.

THE ROLE OF THE PRECODER MATRIX $\mathbf{F}$

Suppose we have two generator matrices $\mathbf{G}$ and $\mathbf{G}' = \mathbf{F}\,\mathbf{G}$ (with corresponding quantities C'_i, θ'_i, and λ'_i). We analyze their MSE's through the relations among their subspaces. To do so we focus on the precoder matrix $\mathbf{F}$. Consider the inverse $\mathbf{F}^{-1}$ of $\mathbf{F}$, which we write as

$$\mathbf{F}^{-1} = [\mathbf{x}_{l_{k-1}}{}^{\mathrm{T}}, ..., \mathbf{x}_{l_0}{}^{\mathrm{T}}]$$

In [4], we show that

$$C'_{2^s} = C_{l_s}$$

Consequently, $\theta'_{2^s} = \theta_{l_s}$, and $\lambda'_{2^s} = \lambda_{l_s}$.

SPECIAL CASE

Suppose that $\mathbf{G}$ does not achieve the lower bound, that is $E_u(\mathbf{G}) > E_u^{lo}$. Then we may be able to achieve E_u^{lo} with $\mathbf{G}' = \mathbf{F}\,\mathbf{G}$. To do so, we need to pick the columns $\mathbf{x}_{l_s}{}^{\mathrm{T}}$ of the precoder inverse $\mathbf{F}^{-1}$ such that $\lambda_{l_s} = \bar{\lambda}_1$ for all s. This is possible if and only if we can find k independent elements, $\mathbf{x}_{l_0}, ..., \mathbf{x}_{l_{k-1}}$, such that $\lambda_{l_s} = \bar{\lambda}_1$ for all s. Then $E_u(\mathbf{G}') = E_u^{lo}$, as desired.

EXAMPLE 1 (CONT.)

For $\mathbf{G} = \mathbf{G}^{cyc}$, we find $\lambda_1 = \lambda_2 = \lambda_8 = \bar{\lambda}_1$, and $\lambda_4 = \bar{\lambda}_2$. Hence

$$\ell = \frac{1}{85}\,(69\,\bar{\lambda}_1 + 16\,\bar{\lambda}_2) = 0.859$$

We check that $\lambda_5 = \bar{\lambda}_1$, and that $(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_5,\mathbf{x}_8)$ are independent. We let

$$\mathbf{F}^{-1} = [\mathbf{x}_8{}^{\mathrm{T}},\mathbf{x}_5{}^{\mathrm{T}},\mathbf{x}_2{}^{\mathrm{T}},\mathbf{x}_1{}^{\mathrm{T}}]$$

It turns out $\mathbf{F} = \mathbf{F}^{-1}$ in this example. The new generator matrix is

$$\mathbf{G}' = \mathbf{F}\,\mathbf{G} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

with $\lambda'_{2^s} = \bar{\lambda}_1$ for all s. Hence $\ell' = \bar{\lambda}_1$, $\mathbf{G}^{min} = \mathbf{G}'$, and $E_u^{min} = E_u(\mathbf{G}') = E_u^{lo}$. In Figure 1, we show $E_t(\mathbf{G}^{cyc})$, E_t^{min}, and E_t^{max}, the best and worst total MSE, respectively. We also show $E_c(\mathbf{G}^{cyc})$, E_c^{min}, and E_c^{max}, the best and worst channel MSE, respectively. Note that the gap between E_t^{min} and E_t^{max} can exceed 2.75 db.

General Case

In general, we may not be able to find k independent elements with $\lambda_{i_s} = \bar{\lambda}_1$. In particular, some codes have $\mu_1 < k$. In this case, it may be necessary to do a full search over the set $\Gamma(C)$ to determine the smallest MSE. The previous analysis suggests a greedy approach to constructing $\mathbf{F}^{-1}$. First take as many independent elements as possible that achieve $\bar{\lambda}_1$. Then take as many elements as possible from that achieve $\bar{\lambda}_2$, while they and the previously chosen elements are independent, and so on.

Example 2

Consider the (5,3;2) code with

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

In Table 2, we list the distinct subspace weight profiles, along with μ_i, $\bar{\theta}_i$, and $\bar{\lambda}_i$ evaluated at $\varepsilon = 0.1$. For the $\mathbf{G}$ above, $\lambda_1 = \lambda_2 = \lambda_4 = \bar{\lambda}_2$, so $\ell = \bar{\lambda}_2$. Since $\mu_1 = 1$ and $\mu_2 = 4$, the best we can hope to get is $\mathbf{G}'$ with $\lambda'_4 = \bar{\lambda}_1$, and $\lambda'_1 = \lambda'_2 = \bar{\lambda}_2$. We check that $\lambda_5 = \bar{\lambda}_1$, and $(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_5)$ are independent. We let

$$\mathbf{F}^{-1} = [\mathbf{x}_5{}^{\mathrm{T}},\mathbf{x}_2{}^{\mathrm{T}},\mathbf{x}_1{}^{\mathrm{T}}]$$

Then $\mathbf{G}' = \mathbf{F}\,\mathbf{G}$ has

$$\ell' = \frac{1}{21}\,(5\,\bar{\lambda}_2 + 16\,\bar{\lambda}_1)$$

Thus $\mathbf{G}^{min} = \mathbf{G}'$, and $E_{min}^{u} > E_{u}^{lo}$.

THE BEST REALIZATION AND THE CHANNEL RAW ERROR RATE

An important issue is whether the order of the distinct probabilities $\bar{\theta}_i$ and distinct eigenvalues $\bar{\lambda}_i$ changes with ε. This in turn affects the choice of $\mathbf{G}^{min}$, the matrix that minimizes the MSE. Except at very large ε, the approximation $\bar{\theta}_i \cong \pi_0 + \bar{a}_{i,d}\,\pi_d$ should be good. This suggests we can rank the $\bar{\theta}_i$'s according to $\bar{a}_{i,d}$, the number of codewords with weight equal to the minimum distance d. For the (7,4;3) and (5,3;2) codes, the order does not change for $0 < \varepsilon < 1/2$. This is also true for several BCH codes and for the Golay codes. Hence $\mathbf{G}^{min}$ is the same for all ε. In general, $\mathbf{G}^{min}$ can change with ε.

EXAMPLE 3

An interesting case is the (15,9;4) cyclic code. It turns out that the order of the critical eigenvalues changes with ε. This results in different generator matrices $\mathbf{G}^{min}$ for small and large ε. In addition, $\mathbf{G}^{min}$ belongs to the special case for small ε, and the general case for large ε. The generator polynomial is

$$g(D) = 1 + D^2 + D^3 + D^4 + D^6$$

Table 3 shows the results at $\varepsilon = 0.1$. There are eleven distinct subspace weight profiles. For $\mathbf{G} = \mathbf{G}^{cyc}$, we have $\lambda_1 = \lambda_2 = \lambda_{128} = \lambda_{256} = \bar{\lambda}_2$, $\lambda_4 = \lambda_{64} = \bar{\lambda}_6$, and $\lambda_8 = \lambda_{16} = \lambda_{32} = \bar{\lambda}_8$. As with the (5,3;2) code, since $\mu_1 = 1$ and $\mu_2 = 15$, the best we can hope to get is $\mathbf{G}'$ with $\lambda'_{256} = \bar{\lambda}_1$, and $\lambda'_1 = \lambda'_2 = \ldots = \lambda'_{128} = \bar{\lambda}_2$. This is satisfied by $\mathbf{G}' = \mathbf{F}\,\mathbf{G}$, where

$$\mathbf{F}^{-1} = \left[\mathbf{x}_{511}{}^{T}, \mathbf{x}_{128}{}^{T}, \mathbf{x}_{93}{}^{T}, \mathbf{x}_{46}{}^{T}, \mathbf{x}_{23}{}^{T}, \mathbf{x}_{11}{}^{T}, \mathbf{x}_{5}{}^{T}, \mathbf{x}_{2}{}^{T}, \mathbf{x}_{1}{}^{T}\right]$$

Hence $\mathbf{G}^{min} = \mathbf{G}'$ belongs to the general case. For large ε, the order of the $\bar{\theta}_i$'s changes. For example, at $\varepsilon = 0.35$,

$$\bar{\theta}_1 = 0.1366 < \bar{\theta}_3 = 0.1369 < \bar{\theta}_2 = 0.1596$$

We purposely avoid reindexing, so we can refer to quantities as they were defined at $\varepsilon = 0.1$. It turns out we can find nine independent elements with $\lambda_i = \bar{\lambda}_2$ (which is now the second largest eigenvalue, after $\bar{\lambda}_0$). The resulting inverse is

$$\mathbf{F}^{-1} = \left[\mathbf{x}_{256}{}^{T}, \mathbf{x}_{128}{}^{T}, \mathbf{x}_{93}{}^{T}, \mathbf{x}_{46}{}^{T}, \mathbf{x}_{23}{}^{T}, \mathbf{x}_{11}{}^{T}, \mathbf{x}_{5}{}^{T}, \mathbf{x}_{2}{}^{T}, \mathbf{x}_{1}{}^{T}\right]$$

which is different from the inverse at $\varepsilon = 0.1$. Hence $\mathbf{G}^{min} = \mathbf{F}\,\mathbf{G}$ is also different, and moreover, it belongs to the special case.

CONCLUSIONS

In this paper, we considered the problem of coupling a quantizer and an error control code. Our analysis focuses on the special case of a uniform source and a uniform quantizer. It can be generalized to other sources, and to equiprobable quantizers (where all bins have the same probability). Preliminary experiments indicate that when applied to images, a well matched quantizer / error control code pair not only gives good MSE results, but also good subjective visual results.

BIBLIOGRAPHY

[1] N. Ahmed and K. R. Rao, Orthogonal transforms for digital signal processing, Springer Verlag (1975).

[2] J. Ashley, "Error correcting codes and quantization," *Submitted to IEEE Transactions on Information Theory* (1993).

[3] R. Blahut, Theory and practice of error control codes, Addison-Wesley (1984).

[4] A. Khayrallah, "On the minimum bit error rate of linear codes," *Submitted to IEEE Transactions on Information Theory* (1994).

[5] S. Lin and D. J. Costello, Error control coding: Fundamentals and applications, Prentice-Hall (1983).

[6] S. McLaughlin, J. Ashley, and D. Neuhoff, "The optimality of the natural binary index assignment," *Submitted to IEEE Transactions on Information Theory* (1993).

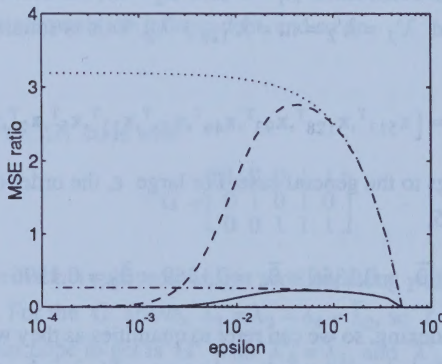


Figure 1. $E_t(\mathbf{G}^{cyc})$ (–), E_t^{max} (--) divided by E_t^{min}, and $E_c(\mathbf{G}^{cyc})$ (-·), E_c^{max} (··) divided by E_c^{min}, for the (7,4;3) code. All ratios in db.

i	0	1	2	3
$\bar{\theta}_i$	1	.933	.914	.869
$\bar{\lambda}_i$	1	.866	.828	.738
μ_i	1	7	7	1

j	a_j	$\bar{a}_{1,j}$	$\bar{a}_{2,j}$	$\bar{a}_{3,j}$
0	1	1	1	1
3	7	4	3	0
4	7	3	3	7
7	1	0	1	0

Table 1. Probabilities, eigenvalues, multiplicities, and weight profiles, for the subspaces of the (7,4;3) Hamming code at $\varepsilon = 0.1$.

i	0	1	2	3
$\bar{\theta}_i$	.608	.605	.599	.592
$\bar{\lambda}_i$	.608	.602	.590	.576
μ_i	1	1	4	2

j	a_j	$\bar{a}_{1,j}$	$\bar{a}_{2,j}$	$\bar{a}_{3,j}$
0	1	1	1	1
2	2	2	1	0
3	4	0	2	2
4	1	1	0	1

Table 2. Probabilities, eigenvalues, multiplicities, and weight profiles, for the subspaces of the (5,3;2) code at $\varepsilon = 0.1$.

i	0	1	2	3	4	5	6	7	8	9	10	11
$\bar{\theta}_i$	.597	.586	.583	.580	.577	.576	.575	.573	.572	.571	.569	.569
$\bar{\lambda}_i$	.597	.576	.569	.563	.557	.555	.553	.550	.547	.546	.542	.542
μ_i	1	1	15	15	30	30	60	60	90	90	60	60

j	a_j	$\bar{a}_{1,j}$	$\bar{a}_{2,j}$	$\bar{a}_{3,j}$	$\bar{a}_{4,j}$	$\bar{a}_{5,j}$	$\bar{a}_{6,j}$	$\bar{a}_{7,j}$	$\bar{a}_{8,j}$	$\bar{a}_{9,j}$	$\bar{a}_{10,j}$	$\bar{a}_{11,j}$
0	1	1	1	1	1	1	1	1	1	1	1	1
4	30	30	22	22	18	18	16	16	14	14	12	12
5	60	0	40	20	32	28	34	26	32	28	30	30
6	60	60	36	36	24	24	30	30	28	28	34	34
7	105	0	56	49	53	52	47	58	48	57	54	51
8	105	105	49	49	53	53	47	47	57	57	51	51
9	60	0	24	36	24	36	30	30	32	28	26	34
10	60	60	20	20	32	32	34	34	28	28	30	30
11	30	0	8	22	18	12	16	14	16	14	18	12
15	1	0	0	1	1	0	1	0	0	1	0	1

Table 3. Probabilities, eigenvalues, multiplicities, and weight profiles, for the subspaces of the (15,9;4) code at $\varepsilon = 0.1$.

On Output Feedback in Variable Structure Control[1]

Michael K.H. Fan
School of Electrical and
Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332

Wen-June Wang
Department of Electrical Engineering
National Central University
Chung Li 320, Taiwan, ROC

Abstract

In this paper, we propose a new variable structure output feedback controller for a class of uncertain systems. Our method only requires the knowledge of upper bounds for the initial state and the uncertainty. It is shown that the proposed controller drives the state trajectory towards a suitable switching hyperplane and forces the trajectory to asymptotically go to the origin along this hyperplane. An upper bound for the time that the state trajectory reaches the switching hyperplane is also computed. Further, it is seen that the state trajectory will be confined in a ball with a computable radius. Numerical examples are provided to illustrate the synthesis of the proposed controller.

1 Introduction

Variable structure control (VSC) is a robust nonlinear control strategy using a discontinuous control signal which changes the system structure upon reaching a certain switching surface [9]. The major merit of the VSC is its insensitivity to the class of uncertainty which satisfies the matching condition [3, 9]. However, many conventional VSC techniques employ full state feedback which in practice is either impossible or infeasible. In fact, the full state feedback is only of academic interest because, by definition, the outputs are the only signals which are accessible. To overcome this difficulty, VSC techniques based on asymptotic observers and dynamic compensators have been proposed [1, 5, 2, 10, 11]. However, these methods usually lead to a controller which significantly increases the complexity of the overall system. Therefore, the static output feedback VSC is worth investigating. Several authors have proposed VSC based on output feedback [7, 2, 4, 6, 13, 12, 8] but the results seem to reply on relatively stronger assumptions.

In this paper, we propose an output feedback VSC to stabilize a class of uncertain systems. The uncertainty is assumed to satisfy a certain cone-type constraint in relation to the states. Also, we assume that an upper bound for the magnitude of the initial state is available. Under the existence of a switching hyperplane, it is shown that the proposed controller will drive the state trajectory towards the switching hyperplane no later than a known time, and then force the trajectory to asymptotically go to the origin in spite of the presence of nonlinearity, disturbance or uncertainty.

[1]This work was performed while the second author, who is supported by the National Science Council in Taiwan, ROC, under the File No. 32004F, is visiting the School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA 30332.

In the sequel, given a vector x, we denote by x^T its transpose, by $\|x\|$ its ℓ_2 or Euclidean norm, and by $\|x\|_1$ its ℓ_1 norm. Given a matrix A, we denote by A^T its transpose and by $\|A\|$ its ℓ_2 induced norm.

2 Problem Formulation

Consider the system described by the following dynamic equations

$$\begin{aligned} \dot{x} &= Ax + f(t,x) + Bu \\ y &= Cx \end{aligned} \tag{2.1}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ the input vector, $y \in \mathbb{R}^p$ the output vector, and the known constant matrices A, B and C are of appropriate dimension. We assume that $m \leq p \leq n$. The function $f(t,x)$ denotes the (possibly time-varying) nonlinearity, disturbance or uncertainty. Also, we make the following assumptions for the problem data:

A.1) There exists a matrix $G \in \mathbb{R}^{m \times p}$ such that GCB is nonsingular and the matrix

$$\left(I - B(GCB)^{-1}GC\right) A$$

is stable in the subspace $\mathcal{H}$ defined by

$$\mathcal{H} = \{x : GCx = 0\} \tag{2.2}$$

That is, let the rows of $V \in \mathbb{R}^{(n-m)\times n}$ form an orthonormal basis of $\mathcal{H}$ (notice that the nonsingularity assumption of GCB implies that GC has full row rank). Then all the eigenvalues of the matrix

$$\hat{A} = V\left(I - B(GCB)^{-1}GC\right)AV^T \tag{2.3}$$

have negative real parts.

A.2) The standard matching condition on $f(t,x)$ is satisfied. That is, for any $t \geq 0$ and $x \in \mathbb{R}^n$, $f(t,x)$ lies in the range space of the matrix B. Further, there exist known scalars $\alpha, \beta \geq 0$ such that for any $t \geq 0$ and $x \in \mathbb{R}^n$, it holds that

$$\|f(t,x)\| \leq \alpha + \beta\|x\| \tag{2.4}$$

Other than these conditions, $f(t,x)$ is unknown.

A.3) The initial state $x(0)$ lies in the ball centered at the origin with a known radius $r > 0$, i.e., $\|x(0)\| \leq r$. Otherwise, $x(0)$ is unknown.

Our goal is to design an output feedback law u such that the closed-loop system under consideration is stable in the sense that $x(t) \to 0$ as $t \to \infty$.

3 Main Result

In this section, we propose a variable structure output feedback controller for the system (2.1). It is shown that the proposed controller drives the state trajectory onto a suitable switching hyperplane and forces the trajectory to asymptotically go to the origin along this hyperplane in spite of the presence of nonlinearity, disturbance or uncertainty. The switching hyperplane is chosen in the form of (2.2) for some $G \in \mathbb{R}^{m\times p}$. Let

$$s = Gy = GCx \tag{3.1}$$

Setting $\dot{s} = 0$ and solving for the equivalent control yield the dynamic equations for the switching hyperplane

$$\dot{x} = \left(I - B(GCB)^{-1}GC\right)Ax \quad \text{and} \quad GCx = 0$$

In view of the assumption A.1 in Section 2, the matrix G can be chosen to have the state trajectory approach to the origin asymptotically provided that we can confine the trajectory to stay within the switching hyperplane. In [9], Utkin shows that the condition $s^T\dot{s} < 0$ for all $x \notin \mathcal{H}$ ensures that the state trajectory is directed towards the switching hyperplane and stays within thereafter once the hyperplane is reached. Hence it remains to synthesize an output feedback control strategy to guarantee that the switching hyperplane can be reached in a finite time.

We shall employ the following form for our control input

$$u = -k(GCB)^{-1}\mathrm{sgn}(s) \tag{3.2}$$

where $\mathrm{sgn}(s)$ denotes the signum function, i.e., the ith component of $\mathrm{sgn}(s) \in \mathbb{R}^m$ is 1 if $s_i > 0$ and -1 if $s_i < 0$. Using (2.1), (3.1) and (3.2), it is easily checked that

$$\begin{aligned} s^T\dot{s} &= s^TGCAx + s^TGCf + s^TGCBu \\ &= s^TGCAx + s^TGCf - k\|s\|_1 \end{aligned}$$

We shall choose k such that

$$s^T\dot{s} \le -\eta\|s\| \tag{3.3}$$

for all t, where η is a positive number to be determined. Also, let us assume for a moment that there exists $r_u > 0$ such that

$$\|x(t)\| \le r_u \tag{3.4}$$

for all t. Later we will show how to compute η and r_u in order to satisfy (3.4). Thus in view of the assumption A.2, it is straight forward to check that the condition $s^T\dot{s} \le -\eta\|s\| < 0$ holds for all $x \notin \mathcal{H}$ if

$$k = \tau_1 + \tau_2 r_u \tag{3.5}$$

where τ_1 and τ_2 are positive numbers defined by

$$\tau_1 = \eta + \alpha\|GC\| \tag{3.6}$$

and

$$\tau_2 = \|GCA\| + \beta\|GC\| \tag{3.7}$$

respectively. Notice that above we have used the inequality $\|s\|_1 \geq \|s\|$.

Now we compute an upper bound for the time that the state trajectory reaches the switching hyperplane. Using (3.3) and the identity

$$s^T\dot{s} = \|s\|\frac{d\|s\|}{dt}$$

we see that if $s \neq 0$ then the magnitude of s will decrease by at least η per unit of time. Therefore, the state trajectory must reach the switching hyperplane no later than

$$T = \frac{\|Gy(0)\|}{\eta} \tag{3.8}$$

Let $\hat{A}$ be defined by (2.3) with an appropriate V. Then it is easy to see that for any $t_1 \geq t_2 \geq T$,

$$\|x(t_1)\| \leq \delta\|x(t_2)\| \tag{3.9}$$

where δ is defined by

$$\delta = \max_{t\geq 0}\left\|e^{\hat{A}t}\right\| \tag{3.10}$$

Note that the "max" in (3.10) is achieved (since $\hat{A}$ is stable) and δ can be determined graphically.

Recall that the state trajectory will move towards the origin once it reaches the switching hyperplane (implied by the assumption A.1). Thus, we can use (3.8) to obtain a bound on how far the trajectory may deviate from the origin. Let $t \in [0, T]$ and define

$$\lambda = \max_{v\in\mathbb{R}^m}\left\{\left\|B(GCB)^{-1}v\right\| : v_i = \pm 1,\ i = 1,\cdots,m\right\} \tag{3.11}$$

Then

$$\begin{aligned}
\|x(t)\| &= \left\|x(0) + \int_0^t (Ax + f + Bu)dt\right\| \\
&\leq \|x(0)\| + \int_0^t \left\|Ax + f - (\tau_1 + \tau_2 r_u)B(GCB)^{-1}\mathrm{sgn}(s)\right\| dt \\
&\leq \|x(0)\| + \int_0^t (\phi_1 + \phi_2\|x\|)dt
\end{aligned}$$

where ϕ_1 and ϕ_2 are positive numbers defined by

$$\phi_1 = \alpha + \tau_1\lambda + \tau_2\lambda r_u \tag{3.12}$$

and

$$\phi_2 = \|A\| + \beta \tag{3.13}$$

respectively. The above inequality on $\|x(t)\|$ implies that for $t \in [0,T]$, $\|x(t)\| \leq z(t)$ where $z(t)$ satisfies the integral equation

$$z(t) = r + \int_0^t (\phi_1 + \phi_2 z(t))dt$$

which is equivalent to the differential equation

$$\dot{z} = \phi_1 + \phi_2 z, \qquad z(0) = r \tag{3.14}$$

It is easy to see that (3.14) has the solution

$$z(t) = \frac{\phi_1}{\phi_2}\left(e^{\phi_2 t} - 1\right) + re^{\phi_2 t}$$

Using (3.9) and (3.10), and since $z(t)$ is monotonically increasing in t, we have

$$\|x(t)\| \leq \delta z(T) = \frac{\delta\phi_1}{\phi_2}\left(e^{\phi_2 T} - 1\right) + \delta re^{\phi_2 T} \tag{3.15}$$

for all t. Notice that the above derivation is valid under the assumption $\|x(t)\| \leq r_u$ for all t. This assumption is ensured if we require that r_u is larger than or equal to the right hand side of (3.15), i.e.,

$$r_u \geq \frac{\delta(\alpha + \tau_1\lambda + \tau_2\lambda r_u)(\phi_3 - 1)}{\phi_2} + \delta r\phi_3 \tag{3.16}$$

where ϕ_3 is defined by

$$\phi_3 = e^{\phi_2 T} \tag{3.17}$$

Thus, the condition (3.16) on η and r_u (since τ_1 and ϕ_3 depend on η) must hold in order to validate the derivation we have obtained so far. It is easy to check that (3.16) has a positive solution for r_u if and only if

$$\frac{\delta\tau_2\lambda(\phi_3 - 1)}{\phi_2} < 1 \tag{3.18}$$

Using (3.8), (3.17) and (3.18), we have the following condition on η for the solvability of (3.16):

$$\eta > \frac{\phi_2\|Gy(0)\|}{\log(\phi_2 + \delta\tau_2\lambda) - \log(\delta\tau_2\lambda)} \tag{3.19}$$

With a fixed η satisfying (3.19), we then solve r_u in (3.16) and obtain

$$r_u \geq \frac{\delta(\alpha + \tau_1\lambda)(\phi_3 - 1) + \delta r\phi_2\phi_3}{\phi_2 - \delta\tau_2\lambda(\phi_3 - 1)} \tag{3.20}$$

The main result of the paper is summarized in the following theorem, whose proof has already been given above.

Theorem 1. Consider the system described by (2.1). Suppose that the assumptions A.1, A.2 and A.3 are satisfied. Let G be any matrix which satisfies the assumption A.1.

Define τ_2, δ, λ and ϕ_2 by (3.7), (3.10), (3.11) and (3.13), respectively. Let η be any positive number which satisfies (3.19). Define τ_1, T and ϕ_3 by (3.6), (3.8) and (3.17), respectively. Also, let r_u be any positive number which satisfies (3.20). Define k by (3.5). Finally, define the control input u by (3.2). Then, the following statements hold: (i) The state trajectory will reach the switching hyperplane $\mathcal{H}$ defined by (2.2) no later than the time T. (ii) The state trajectory satisfies the inequality $\|x(t)\| \leq r_u$ for all t. (iii) The state trajectory will go to the origin asymptotically as the time approaches the infinity. □

There is still some freedom left in choosing the value of k (by choosing different values of η and r_u in (3.19) and (3.20)). We may use the freedom to minimize k which will in turn reduce the chattering behavior in the practical situation. First, it is clear that for a fixed η, we should choose r_u in (3.20) using the inequality as an equality. Thus r_u is uniquely determined by η and we can view that k solely depends on η. Second, it can be easily checked that k becomes large when η is chosen too large or too small (of course (3.19) needs to be satisfied). That is to say, there exists a finite η which leads to the smallest k among those obtained using Theorem 1. The optimal η then can be obtained graphically using the plot of k versus η. We will illustrate this procedure in the next section.

4 Examples

4.1 Example 1

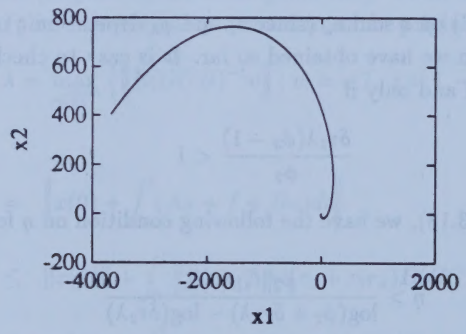


Figure 1: Phase-plane trajectory without input

Consider the system described by the following nonlinear dynamic equations

$$\begin{aligned} \dot{x}_1 &= 2x_1 - 3x_2 \\ \dot{x}_2 &= x_1 + 2x_2 + \sin(x_1) + \cos(t)\log(|x_2| + 1) + u \\ y &= -x_1 + x_2 \end{aligned}$$

with the initial state $(0.1, 0)$. The above dynamic equations can be easily transformed into the form (2.1) with the following problem data:

$$A = \begin{bmatrix} 2 & -3 \\ 1 & 2 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = [-1 \quad 1], \quad f(t,x) = \begin{bmatrix} 0 \\ \sin(x_1) + \cos(t)\log(|x_2|+1) \end{bmatrix}$$

It is easy to see that the assumption A.1 is satisfied with $G = 1$ (or any nonzero value). Also, the matching condition on $f(t,x)$ is satisfied, and for any $t \geq 0$ and $x \in \mathbb{R}^2$, (2.4) holds with $\alpha = \beta = 1$ (we have used the inequality $\log(a+1) \leq a$ for $a \geq 0$ in obtaining β). Finally, the assumption A.3 holds if we choose $r = 0.1$.

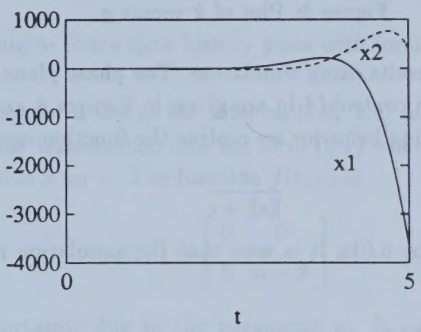


Figure 2: State time history plots without input

The system is unstable with no input. This can be seen from the phase plane trajectory in Figure 1 or the state time history plots in Figure 2. To carry out the proposed method in designing a variable structure output feedback controller, we first compute τ_2, δ, λ and ϕ_2 using (3.7), (3.10), (3.11) and (3.13), respectively:

$$\tau_2 = 6.513, \qquad \delta = 1, \qquad \lambda = 1, \qquad \phi_2 = 4.828$$

Then we obtain 0.871 for the right hand side of (3.19). Second, we set η equal to each of a set of numbers larger than 0.871 and then correspondingly compute τ_1, T, ϕ_3 and k using (3.6), (3.8), (3.17) and (3.5), respectively. The plot of k versus η is given in Figure 3. It shows that the optimal η is about 2.58. The corresponding T, r_u and k are then

$$T = 0.0388, \qquad r_u = 0.4616, \qquad k = 7.001$$

Therefore, we have obtained a control input

$$u = -k(GCB)^{-1}\mathrm{sgn}(s) = -7.001\ \mathrm{sgn}(y) \tag{4.1}$$

which, according to Theorem 1, will drive the state trajectory to hit the hyperplane $\{x : x_1 = x_2\}$ within 0.0388 second and then force the trajectory to asymptotically go to the origin along this hyperplane. Also the magnitude of the state vector will be no more than 0.4616 for all times.

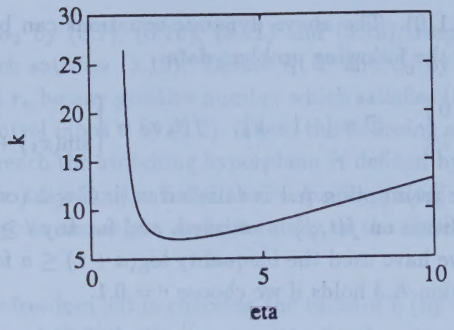


Figure 3: Plot of k versus η

Let us also verify the results using simulation. The phase plane trajectory and state time history plots with the control (4.1) are given in Figures 4 and 5, respectively. In order to reduce the chattering behavior we replace the function $\text{sgn}(s)$ by

$$\frac{s}{\|s\| + \epsilon} \tag{4.2}$$

where ϵ is chosen here to be 0.01. It is seen that the simulation results are consistent with Theorem 1.

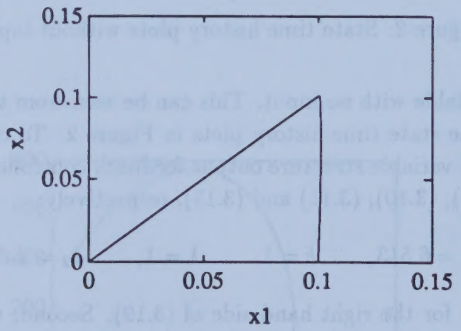


Figure 4: Phase-plane trajectory with input (4.1)

4.2 Example 2

In Example 1, we have used the function $f(t,x)$ to represent the nonlinearity in the system. In this example, we show that $f(t,x)$ can also be used to represent the uncertainty due to system parameter variation. For simplicity, we consider the following linear system:

$$\begin{aligned} \dot{x}_1 &= 2x_1 - 3x_2 \\ \dot{x}_2 &= x_1 + ax_2 + u \\ y &= -x_1 + x_2 \end{aligned}$$

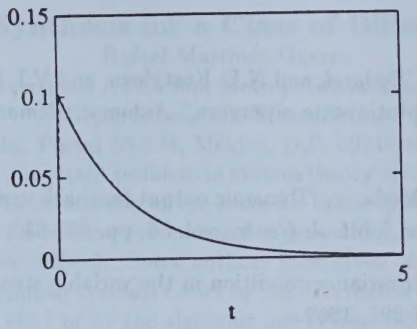


Figure 5: State time history plots with input (4.1)

Suppose that the parameter a lies in the interval [1.9, 2.1] and otherwise is unknown. The system can be easily transformed into the form (2.1) where the matrix A is defined using the (nominal) value 2 for a. The function $f(t,x)$ is

$$f(t,x) = \begin{bmatrix} 0 & 0 \\ 0 & a-2 \end{bmatrix} x$$

which reflects the uncertainty due to the parameter a. It can be easily checked that $f(t,x)$ satisfies both the matching condition and (2.4) with $\alpha = 0$ and $\beta = 0.1$. As in Example 1, the assumption A.1 is satisfied with $G = 1$. We carry out the proposed method on this problem and obtain the optimal $\eta = 1.63$ which leads to the following control law:

$$u = -3.609 \ \mathrm{sgn}(y)$$

The resulting closed-loop phase plane trajectory with $a = 2.1$ is shown in Figure 6. Again, (4.2) with the same ϵ replaces the function $\mathrm{sgn}(s)$.

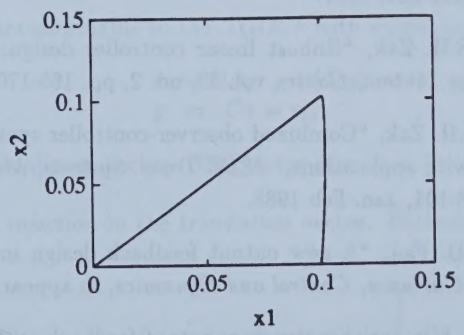


Figure 6: Phase plane trajectory for Example 2

References

[1] A.G.S. Bondaref, S.A. Bodaref, and N.E. Kostyleva, and V.I. Utkin, "Sliding modes in systems with asymptotic state observers," *Automat. Remote Control*, vol. 46, pp. 679-684, 1985.

[2] B.M. Diong and J.V. Medanic, "Dynamic output feedback variable structure control for system stabilization," *Int. J. Contr.*, vol. 56, pp. 607-630, 1992.

[3] B. Drazenovic, "The invariance condition in the variable structure systems," *Automatica*, vol. 5, pp. 287-295, 1969.

[4] R. El-Khazali and R.A. DeCarlo, "Variable structure output feedback control," *Proc. of American Control Conf.*, pp. 871-875, Chicago, IL, 1992.

[5] S.V. Emelyanov, S.K. Korovin, A.L. Nersisian and Y.E. Nisenzon, "Output feedback stabilization of uncertain plants: a variable structure systems approach," *Int. J. Contr.*, vol. 55, pp. 61-81, 1992.

[6] R. El-Khazali and R.A. DeCarlo, "Output feedback variable structure control design using dynamic compensator for linear systems," *Proc. American Control Conf.*, pp. 954-958, San Francisco, CA, 1993.

[7] B.S. Heck and A.A. Ferri, "Application of output feedback variable structure systems," *AIAA J. Guidance, Control, and Dynamics*, vol. 12, pp. 932-935, 1989.

[8] B.S. Heck, S.V. Yallapragada and M.K.H. Fan, "Numerical methods to design the reaching phase of output feedback variable structure control," *Automatica*, to appear.

[9] V.I. Utkin, "Variable structure systems with sliding modes," *IEEE Trans. Automat. Contr.*, vol. 22, pp. 212-222, 1977.

[10] B.L. Walcott and S.H. Żak, "Robust linear controller design: time domain approach," *IEEE Trans. Autom. Contr.*, vol. 32, no. 2, pp. 166-170, 1987.

[11] B.L. Walcott and S.H. Żak, "Combined observer-controller synthesis for uncertain dynamical systems with applications," *IEEE Trans. Systems, Man and Cybernetics*, vol. 10, no. 1, pp. 88-104, Jan.-Feb 1988.

[12] W.J. Wang and Y.D. Fan, "A new output feedback design in variable structure systems," *AIAA J. Guidance, Control and Dynamics*, to appear.

[13] S.H. Żak and S. Hui, "On variable structure output feedback controller for uncertain dynamic systems," *IEEE Trans. Autom. Contr.*, vol. 38, no. 10, pp. 1509-1512, 1993.

Observer Synthesis for a Class of Bilinear Systems.

Rafael Martínez-Guerra
Universidad Autónoma Metropolitana-Iztapalapa
Departamento de Matemáticas
Apdo. Postal 55-534, México, D.F. 09340 MEXICO

1.INTRODUCTION. An important problem in system theory is to estimate the non-directly accessible quantities. An estimation of the state for linear systems is given by the Luenberger's observer while for bilinear systems and in general for nonlinear ones the construction of observers is more difficult. Some authors have given some ideas for the synthesis or observers design of bilinear systems either by the differential geometric viewpoint (Hermann, Krener, Isidori, etc.) or by the algebraic one (Fliess 1986). The latter approach is based on differential algebra. In this paper the observer synthesis problem is addressed in the algebraic direction.

2.AN ASYMPTOTIC OBSERVER FOR A CLASS OF BILINEAR SYSTEMS. Let us now, consider the following Bilinear *System Class* (BS):

$$\begin{aligned} \dot{x} &= (A_0 + \sum_i^m A_i u_i)x + Bu, \quad x \in \mathbf{R}^n \\ y &= Cx + Du, \qquad\qquad\quad y \in \mathbf{R} \end{aligned} \tag{1}$$

where A_i, $0 \leq i \leq m$, B, C, and D are real matrices of appropriate size.

Suppose that BS (1) is observable in the Diop-Fliess'Algebraic Observability sense [1,2], then let us pose the following problem.

Is it possible to synthesize an asymptotic observer for the system (1)?

The following proposition states that by an appropriate choice of the differential primitive element [1] it is possible to obtain a *Translated* Fliess' GOCF *(TGOCF)* with *output injection.*

Proposition 1. Let the BS (1) be given. If the differential primitive element is chosen like:

$$y = \sum_i^n \alpha_i x_i + \sum_j^m \beta_j u_j \qquad \alpha_i, \beta_j \in k\langle u \rangle$$

then the BS (1) is transformable to the *TGOCF* with *output injection* given by:

$$\begin{aligned} \dot{\tau} &= A_u \tau + \varphi(u, y) \\ y &= C\tau = \tau_1, \end{aligned} \tag{2}$$

where $\varphi(u, y)$ is a nonlinear vector. The vector $\varphi(u, y)$ is the translation vector of the GOCF and y
denotes the *output injection* in the translation vector. Futhermore, the matrix A_u has entries in $k\langle u \rangle$.

$$A_u = \begin{bmatrix} 0 & I \\ 0 & -\sigma(u, u^{(1)}, \ldots, u^{(\gamma)}) \end{bmatrix}, \quad \varphi(u, y) = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ g(u, \ldots, u^{(\gamma)}, y) \end{bmatrix}$$

Remark 1. We can take the $BS(1)$ into a *TGOCF* by means of $\tau = \theta x$ (3)
where θ is a matrix with entries in $k\langle u \rangle$.

Lemma 1. Consider the observable single output system (2). Then

$$\dot{\hat{\tau}} = A_u\hat{\tau} - L_u(y - \hat{y}) + \varphi(u, y) \tag{4}$$

is an asymptotic observer, where $\hat{\tau}$ is an estimate of τ. Furthermore, the entries of observer's gain matrix L_u are in $k\langle u\rangle$ and $\|\tau(t) - \hat{\tau}(t)\| \leq \nu \exp(-\delta t)$ with constant $\delta > 0$, δ independent of the input.

Corollary 1. Suppose that u is a *good input* [4]. Then, the dynamical system (4) along with: $x = \theta^{-1}\tau$ (5) constitute an asymptotic observer for the BS class (1).

3. APPLICATION TO A CHEMICAL REACTOR MODEL. The following two-dimensional single-input single-output system represents a chemical reactor model [4]

$$\begin{aligned} \dot{x}_1 &= u(C_e - x_1) - rx_1 \\ \dot{x}_2 &= rx_1 - ux_2 \\ y &= x_2 + u \end{aligned} \tag{6}$$

with coefficients in the field of real numbers $\mathbf{R}$, which is considered as an ordinary differential field. In (6) x_1 and x_2 denote the reactant and product concentrations respectively. The input u corresponds to the input-flow of reactant r and C_e denote kinetic and reactor parameters.

If u, and $y = x_2 + u$ are measured then x_1 and x_2 verify: $x_2 - y + u = 0$ and $rx_1 - \dot{y} - uy + u^2 + \dot{u} = 0$. Hence $k\{u, y, x\}$ is integral over $k\{u, y\}$, and x_1 and x_2 are universally observable [2,3].

The system (6) is carried out into the *TGOCF* (2) with output injection by means of the differential primitive element $z_1 = x_2 + u$. Where A_u and $\varphi(u, y)$ are given respectively by,

$$A_u = \begin{bmatrix} 0 & I \\ 0 & -(2u + r) \end{bmatrix}, \varphi(u, y) = \begin{bmatrix} 0 \\ ruC_e + [-u^2 - \dot{u} - ur]y + u^3 + 3u\ddot{u} + ru^2 + r\dot{u} + \ddot{u} \end{bmatrix}$$

From lemma 1, it is possible to construct an asymptotic observer (4) for system (6). This is attained by choosing the observer's gain matrix $L_u = [L_{u_1}, L_{u_2}]^T$ in the form:

$$\begin{aligned} L_{u_1} &= \psi_1 + (2u + r), \quad \psi_1 < 0 \\ L_{u_2} &= -\psi_2 - (\psi_1 + 2u + r)(2u + r), \quad \psi_2 > 0. \end{aligned}$$

Remark 2. The observer (4)-(5) has been tested numerically in the case of a chemical reactor. Applying an arbitrary observer pole placement an exponential decaying in the observation error is obtained.

4. REFERENCES.

[1] Diop, S. and Fliess, M., "*Nonlinear observability, identifiability, and persistent trajectories*". IEEE Proc. 30th. Conf. Dec. Contr., Brighton, England, December 1991, pp. 714-719.

[2] Diop, S., "*On Universal Observability*". IEEE Proc. 31st. Conf. Dec. Contr., Tucson, Az., Dec. 1992, pp. 3669-3672.

[3] Diop, S. "*Closedness of morphisms of differential algebraic sets. Applications to system theory*", Forum Math., to appear.

[4] Martínez, R. and Alvarez, J., "*On Observers for a Class of Bilinear Systems*", IEEE Proc. 2nd. Conf. on Contr. Appl. Vancouver, Canada, Sept. 1993, pp. 777-781.

Self-Optimizing Adaptive Policies for A Class of Partially Observed Controlled Markov Chains

Younseok Choo
Quality Assurance Section, ETRI
161 Kajeong-dong Yusong-gu, Taejon, Korea 305-350
Aristotle Arapostathis
Department of Electrical and Computer Engineering
The University of Texas at Austin
Austin, TX 78712-1084, USA

Abstract

In typical adaptive control problems, one important question is whether an adaptive policy is self-optimizing, i.e., whether the cost incurred by the adaptive policy converges to the one achieved by optimal policy corresponding to the true parameter. In this paper we are concerned with the adaptive control of Partially Observed Controlled Markov Chains (POCMCs) with finite states and actions. Some sufficient conditions are presented for the adaptive policies to be self-optimizing.

1 POCMC

Let $\{x_n\}$ be a sequence of random variables defined on a probability space $(\Omega, \mathcal{F}, \mathcal{P})$ denoting the states of a system. For each n, x_n takes values in the finite state space $X = \{0, \cdots, N_X\}$. The control u_n takes values from the finite action space $U = \{0, \cdots, N_U\}$. The state dynamics are governed by a Markovian law. The state transition matrix is denoted by $P(u) = [p_{ij}(u)]$, where

$$p_{ij}(u) = \mathcal{P}[x_{n+1} = j \mid x_n = i, u_n = u].$$

At each time n, the state x_n is observed through a noise-corrupted random variable y_n, where y_n takes values in the finite observation space $Y = \{0, \cdots, N_Y\}$. The observation matrix is given by $Q(u) = [q_{ij}(u)]$, where

$$q_{ij}(u) = \mathcal{P}[y_{n+1} = j \mid x_{n+1} = i, u_n = u].$$

*This work was supported in part by the Texas Advanced Research Program (Advanced Technology Program) under Grant No. 003658-186, in part by the Air Force Office of Scientific Research under Grant F49620-92-J-0045 and in part by the Post Doc. Program of ETRI, Korea.

Let $c(\cdot,\cdot)$ denote the bounded one-step cost. The sequence of events is assumed as follows; at time epoch n the system is in state x_n, observation y_n becomes available, control u_n is applied, and cost $c(x_n, u_n)$ is incurred; transition to a state x_{n+1} has occurred, another observation y_{n+1} becomes available, a new control u_{n+1} is applied, and then another cost $c(x_{n+1}, u_{n+1})$ is incurred; and so on.

Let Δ be the set of all probability distributions on X, i.e.,

$$\Delta = \{p \in \Re^{N_X} : \sum_{i=0}^{N_X} p^{(i)} = 1, \quad p^{(i)} \geq 0\} \tag{1.1}$$

where $p = (p^{(1)}, \cdots, p^{(N_X)})$. To each initial state x_0 and an (admissible) policy μ, we associate the average cost

$$J(\mu, x_0) = \limsup_{n\to\infty} E^{\mu}_{x_0}[\frac{1}{n}\sum_{i=0}^{n-1} c(x_i, u_i)] \tag{1.2}$$

where expectation is taken with respect to the unique probability measure $\mathcal{P}^{\mu}_{x_0}$ on $\Delta \times (U \times Y)^{\infty}$ induced by x_0 and μ [3]. The average cost optimal control problem is that of finding an optimal policy which minimizes the cost given in (1.2). The corresponding cost function is the (average) optimal cost function.

Consider the conditional probability distribution given by

$$p^{(i)}_n = \mathcal{P}^{\mu}_{p_0}[x_n = i \mid y_n, u_{n-1}], \quad n \geq 1$$

with $p^{(i)}_0 = \mathcal{P}[x_0 = i]$. As is well known, the problem with partial observations can be transformed into an equivalent problem with complete observations [1], [3]. In the latter problem, then, the new state becomes a probability distribution, sometimes called a sufficient statistic. In the present context, p_n becomes a sufficient statistic with the state space Δ. The problem defined above can be restated as that of finding an optimal policy which minimizes (using the same notation J)

$$J(\mu, p_0) = \limsup_{n\to\infty} E^{\mu}_{p_0}[\frac{1}{n}\sum_{i=0}^{n-1} \overline{c}(p_i, u_i)] \tag{1.3}$$

where $\overline{c}(p, u) = \sum_{i=0}^{N_X} p^{(i)} c(i, u)$. The (average) optimal cost function is denoted by $J_*(\cdot)$.

Let $\overline{Q}_y(u) = diag\{q_{iy}(u)\}$ and let 1 denote the N_X-dimensional column vector. Then, using Bayes' rule, p_n can be computed recursively as follows [1], [4]:

$$p_{n+1} = \sum_{y\in Y} T(y, p_n, u_n) \cdot I[y_{n+1} = y] \tag{1.4}$$

where

$$V(y, p, u) = 1'\overline{Q}_y(u)P'(u)p \tag{1.5}$$

$$T(y, p, u) = \frac{\overline{Q}_y(u)P'(u)p}{V(y, p, u)}. \tag{1.6}$$

Here $V(y,p,u)$ is the one-step ahead conditional probability of the observation being y given the control u and the *a priori* probability p of the system. Similarly, $T(y,p,u)$ is the *a posteriori* conditional probability of the system given that control u was applied, observation y obtained and the *a priori* probability p.

A well-known property is that $\{p_n\}$ is a controlled Markov chain with transition probability [1]

$$\mathcal{K}(B \mid p,u) = \sum_{y \in Y} V(y,p,u) \cdot I[T(y,p,u) \in B].$$

Let Γ be a constant and h a real-valued function defined on Δ. Then the pair (Γ, h) is said to be a solution to the Average Cost Optimality Equation (ACOE) if, for all $p \in \Delta$

$$\Gamma + h(p) = \min_{u \in U} \{\overline{c}(p,u) + \sum_{y \in Y} V(y,p,u) h(T(y,p,u))\}. \tag{1.7}$$

The following theorem is very standard in the theory of controlled Markov chains [3].

Theorem 1 *Suppose (Γ, h) is a solution to the ACOE, where Γ is a constant and h is a real-valued bounded function on Δ. Then:*
(i) $J_*(p) = \Gamma$ for all $p \in \Delta$.
(ii) If μ^* is a stationary policy such that $\mu^*(p)$ minimizes the right-hand side of the ACOE, i.e., for all $p \in \Delta$

$$\Gamma + h(p) = \overline{c}(p, \mu^*(p)) + \sum_{y \in Y} V(y,p,\mu^*(p)) h(T(y,p,\mu^*(p))),$$

then μ^* is optimal and $J(\mu^*, p) = \Gamma$ for all $p \in \Delta$.

2 Self-Optimization of Adaptive Policies

Let Θ be a compact parameter set and parameterize by $\theta \in \Theta$ a class of POCMCs as described in the previous section. Therefore we use the expressions $P(u,\theta)$, $Q(u,\theta)$, $T(y,p,u,\theta)$ and $V(y,p,u,\theta)$ to denote the dependence of related matrices, probabilities on θ.

For a moment assume that, for each $\theta \in \Theta$, the ACOE has a bounded solution $(\Gamma_\theta, h_\theta(\cdot))$ with $\Gamma_\theta \in \Re$, i.e., for all $p \in \Delta$, $\theta \in \Theta$

$$\Gamma_\theta + h_\theta(p) = \min_{u \in U} \{\overline{c}(p,u) + \sum_{y \in Y} V(y,p,u,\theta) h_\theta(T(y,p,u,\theta))\}. \tag{2.8}$$

Then a family of stationary optimal policies $\{\mu(\cdot,\theta)\}_{\theta \in \Theta}$ is available for decision-making by Theorem 1.

Fix a POCMC with a parameter $\theta^* \in \Theta$. If θ^* is unknown, then we have an adaptive control problem. Let $\{\hat{\theta}_n\}$ be a sequence of estimates of the unknown parameter θ^*. At this time no assumption is made on the type of estimator except that, for each n, $\hat{\theta}_n$

is assumed to have some required measurability. Assuming the certainty equivalence control is adopted, $\{\hat{p}_n\}$ is computed recursively by

$$\hat{p}_{n+1} = \sum_{y \in Y} V(y, \hat{p}_n, u_n, \hat{\theta}_n) \cdot I[y_{n+1} = y] \tag{2.9}$$

where

$$u_n = \mu(\hat{p}_n, \hat{\theta}_n). \tag{2.10}$$

Let μ^a denote the adaptive policy given by (2.9), (2.10).

Define the history spaces $\{H_n\}$ by

$$\begin{aligned} H_0 &= Y, \\ H_n &= H_{n-1} \times U \times Y, \quad n \geq 1. \end{aligned}$$

For each $\epsilon > 0$, define a compact subset Δ_ϵ of Δ by

$$\Delta_\epsilon = \Delta \cap \{p^{(i)} \geq \epsilon, \quad 0 \leq i \leq N_X\}.$$

We use the following assumptions.

(A1) $\hat{\theta}_n$ is $\sigma(H_n)$-measurable and $\hat{\theta}_n \to \theta^*$ in probability under $\mathcal{P}_{p_0}^{\mu^a}$ for all $p_0 \in \Delta$.

(A2) $\hat{p}_n - p_n \to 0$ in probability under $\mathcal{P}_{p_0}^{\mu^a}$ for all $p_0 \in \Delta$.

(A3) $P(u, \theta)$ and $Q(u, \theta)$ are continuous in θ for each $u \in U$.

(A4) For each $\theta \in \Theta$, the ACOE has a unique solution $(\Gamma_\theta, h_\theta(\cdot))$ such that

(i) $h_\theta(\cdot)$ is concave,

(ii) Γ_θ and $h_\theta(p)$ are bounded on Θ and $\Theta \times \Delta$, respectively.

(iii) for some fixed $\overline{p} \in \Delta$, $h_\theta(\overline{p}) = 0$ for all $\theta \in \Theta$.

(A5) There exist $\epsilon > 0$, $k \geq 1$ such that

$$p_0 \in \partial\Delta \Rightarrow p_k \in \Delta_\epsilon \text{ uniformly in } \theta.$$

In [2] it is shown that the adaptive policy μ^a is self-optimizing under (A1)-(A3) and the continuity of $h_\theta(\theta)$. In fact (A4) and (A5) are used to guarantee the latter property. In (A4), (i) is not difficult to verify since $h_\theta(\cdot)$ is usually expressed as a limit of the differential discounted costs. In many cases, (ii) is also easy to prove using the boundedness of one-step cost. The uniqueness and (iii) are satisfied if $(\Gamma_\theta, h_\theta(\cdot))$ is a unique solution to the ACOE up to the constant. (A5) implies there exists an invariant subset Δ_ϵ of Δ. For simplicity, assume the observation matrix Q depends on u only and that $q_{ij} > 0$ for all $i \in X$, $j \in Y$. Then, a simple sufficient condition for (A5) is that $P(u, \theta)$ is positive for each $u \in U$. More generally, (A5) is satisfied if there is $k \geq 1$ such that for each $\{u_n\}_{n=1}^k$

$$P_k(\theta) := P(u_1, \theta) \cdots P(u_k, \theta) \tag{2.11}$$

is continuous and positive. Assumption (A5) will be weakened later. Define

$$\Delta_\epsilon(1) := \{p \in \Delta : p_0 = p \Rightarrow p_1 \in \Delta_\epsilon \text{ uniformly in } \theta\}$$

and for $2 \leq i \leq k$

$$\Delta_\epsilon(i) := \{p \in \Delta : p_0 = p \Rightarrow p_1 \in \Delta_\epsilon(i-1) \text{ uniformly in } \theta\}.$$

Theorem 2 *Assume (A1)-(A5) hold. Then the adaptive policy given by (2.9), (2.10) is self-optimizing.*

Proof: We show that $h_\theta(p)$ is continuous in (θ, p) under (A3)-(A5). Then the theorem follows from [2]. Let $\theta_n \to \overline{\theta}$. Since $\{\Gamma_{\theta_n}\}$ is bounded, there exist a subsequence (again indexed by n) $\{\Gamma_{\theta_n}\}$ and a real number $\overline{\Gamma}$ such that $\Gamma_{\theta_n} \to \overline{\Gamma}$. By (A4)-(ii) and [5, Theorem 10.6], the family of functions $\{h_{\theta_n}(\cdot)\}$ is equi-Lipschitzian on Δ_ϵ. From the ACOE, $\{h_{\theta_n}(\cdot)\}$ is equi-Lipschitzian on $\Delta_\epsilon(1)$. Continuing the same argument, we have that $\{h_{\theta_n}(\cdot)\}$ is equi-Lipschitzian on $\Delta_\epsilon(k)$. But $\Delta_\epsilon(k) = \Delta$ by (A5). Then, by the Arzela-Ascoli theorem, there is a further subsequence (again indexed by n) $\{h_{\theta_n}(\cdot)\}$ and a continuous function $\overline{h}(\cdot)$ such that $h_{\theta_n}(\cdot) \to \overline{h}(\cdot)$ uniformly on Δ. Taking the limit in the ACOE for the sequence $\{\theta_n\}$, we have, for all $p \in \Delta$

$$\overline{\Gamma} + \overline{h}(p) = \min_{u \in U}\{p'C + \sum_{y \in Y} V(y, p, u, \overline{\theta})\overline{h}(T(y, p, u, \overline{\theta}))\}.$$

Clearly $\overline{h}(\overline{p}) = 0$ and $\overline{h}(\cdot)$ is concave. By the uniqueness, $\overline{\Gamma} = \Gamma_{\overline{\theta}}$ and $\overline{h}(\cdot) = h_{\overline{\theta}}(\cdot)$, which implies that $h_\theta(p)$ is continuous in θ for each $p \in \Delta$. Then, by [5, Theorem 10.7], $h_\theta(p)$ is continuous in (θ, p) on $\Theta \times \Delta^o$. Let $(\theta, p_0) \in \Theta \times \partial\Delta$. Then for a sequence $(u_0, y_1), (u_1, y_2), \cdots, (u_{k-1}, y_k)$

$$(\theta, p_k) \in \Theta \times \Delta_\epsilon.$$

By the continuity of $T(y, p, u, \theta)$ and $h_\theta(p)$ on $\Theta \times \Delta^o$, $h_\theta(p)$ is also continuous at (θ, p_0). **Q.E.D.**

Now we consider a weaker version of (A5).

(A6) There exist $\epsilon > 0$, $k_0 \geq 1$ such that

$$p_0 \in \partial\Delta \Rightarrow p_k \in \Delta_\epsilon \text{ for some } k \leq k_0 \text{ uniformly in } \theta.$$

Clearly (A5) is stronger that (A6). Under the same condition on Q, (A6) holds if there is $k_0 \geq 1$ such that, for each $\{u_n\}_{n=1}^{k_0}$, $P_k(\theta)$ in (2.11) is continuous and positive for some $k \leq k_0$. A condition equivalent to (A6) is stated in the following lemma.

Lemma 1 *(A6) is equivalent to: there exist $\epsilon > 0$, $k_0 \geq 1$ such that*

$$p_0 \notin \Delta_\epsilon \Rightarrow p_k \in \Delta \text{ for some } k \leq k_0 \text{ uniformly in } \theta.$$

Proof: (i) $\Leftarrow$ is trivial. (ii) To show $\Rightarrow$, suppose not. Fix a sequence $\{\epsilon_n\}$ such that $\epsilon_n \downarrow 0$, $\epsilon_1 < \epsilon$. Then for each n, there exist $p_0^n \notin \Delta_{\epsilon_n}$, θ_n and a sequence $(u_0^n, y_1^n), \cdots, (u_{k_0-1}^n, y_{k_0}^n)$, such that $p_k \notin \Delta_{\epsilon_n}$ for all $k \leq k_0$. Furthermore we can assume that

$$(u_0^n, y_1^n), \cdots, (u_{k_0-1}^n, y_{k_0}^n) = (u_0', y_1'), \cdots, (u_{k_0-1}', y_{k_0}')$$

for all n since U and Y are finite. Since Δ and Θ are compact, there is a subsequence (again indexed by n) $\{p_0^n, \theta_n\}$ such that

$$(p_0^n, \theta_n) \to (p_0', \theta) \in \Theta \times \Delta.$$

In fact $p_0' \in \partial\Delta$. Now we show $p_k' \notin \Delta_\epsilon$ for all $k \leq k_0$. Then we have a contradiction and the proof is complete. By the continuity of the mapping $(\theta, p) \mapsto T(y, p, u, \theta)$ for each y, u, we have

$$p_1^n = T(y_1', p_0^n, u_0', \theta_n) \to T(y_1', p_0', u_0', \theta) = p_1'$$

and for $2 \leq k \leq k_0$

$$p_k^n = T(y_1', p_{k-1}^n, u_0', \theta_n) \to T(y_1', p_{k-1}', u_0', \theta) = p_k'.$$

For sufficiently large N, $p_k^n \notin \Delta_{\epsilon/2}$ for all $n \geq N$, $k \leq k_0$, which implies $p_k' \notin \Delta_\epsilon$ for all $k \leq k_0$ (contradiction). **Q.E.D.**

For each $1 \leq i \leq k_0$, define

$$\Delta_\epsilon(i) = \{p \notin \Delta_\epsilon : p_0 = p \Rightarrow p_i \in \Delta_\epsilon \text{ uniformly in } \theta\}$$

which is possible by Lemma 1. Then for $2 \leq i \leq k_0$

$$p_0 \in \Delta_\epsilon(i) \Rightarrow p_1 \in \Delta_\epsilon \cup \Delta_\epsilon(i-1).$$

Theorem 3 *Under (A1)-(A4) and (A6), the conclusion of Theorem 2 continues to hold.*

Proof: Again, we show the continuity of $h_\theta(\theta)$ under (A3), (A4) and (A6). Let $\theta_n \to \overline{\theta}$. Since $\{\Gamma_{\theta_n}\}$ is bounded, there exist a subsequence (again indexed by n) $\{\Gamma_{\theta_n}\}$ and a real number $\overline{\Gamma}$ such that $\Gamma_{\theta_n} \to \overline{\Gamma}$. By [5, Theorem 10.6], the family of functions $\{h_{\theta_n}(\cdot)\}$ is equi-Lipschitzian on Δ_ϵ. Then, by the Arzela-Ascoli theorem, there is a further subsequence (again indexed by n) $\{h_{\theta_n}(\cdot)\}$ and a continuous function $h_\epsilon(\cdot)$ such that $h_{\theta_n}(\cdot) \to h_\epsilon(\cdot)$ uniformly on Δ_ϵ.

Assume $p \in \Delta_\epsilon(1)$. Taking limit in the ACOE for the sequence $\{\theta_n\}$, we have

$$\overline{\Gamma} + \lim_{n\to\infty} h_{\theta_n}(p) = \min_{u\in U}\{p'C + \sum_{y\in Y} V(y, p, u, \overline{\theta}) h_\epsilon(T(y, p, u, \overline{\theta}))\}.$$

Define

$$h_1(p) = \begin{cases} h_\epsilon(p) & \text{for } p \in \Delta_\epsilon, \\ \lim_{n\to\infty} h_{\theta_n}(p) & \text{for } p \in \Delta_\epsilon(1). \end{cases}$$

Then for $p \in \Delta_\epsilon(1)$

$$\overline{\Gamma} + h_1(p) = \min_{u\in U}\{p'C + \sum_{y\in Y} V(y,p,u,\overline{\theta})h_1(T(y,p,u,\overline{\theta}))\}.$$

Note that $h_{\theta_n}(\cdot) \to h_1(\cdot)$ uniformly on $\Delta_\epsilon \cup \Delta_\epsilon(1)$.

Assume $p \in \Delta_\epsilon(2)$. Since $p_0 \in \Delta_\epsilon(2) \Rightarrow p_1 \in \Delta_\epsilon \cup \Delta_\epsilon(1)$, taking limits in the ACOE for the sequence $\{\theta_n\}$

$$\overline{\Gamma} + \lim_{n\to\infty} h_{\theta_n}(p) = \min_{u\in U}\{p'C + \sum_{y\in Y} V(y,p,u,\overline{\theta})h_1(T(y,p,u,\overline{\theta}))\}.$$

Define

$$h_2(p) = \begin{cases} h_1(p) & \text{for } p \in \Delta_\epsilon \cup \Delta_\epsilon(1), \\ \lim_{n\to\infty} h_{\theta_n}(p) & \text{for } p \in \Delta_\epsilon(2). \end{cases}$$

Then for $p \in \Delta_\epsilon(2)$

$$\overline{\Gamma} + h_2(p) = \min_{u\in U}\{p'C + \sum_{y\in Y} V(y,p,u,\overline{\theta})h_2(T(y,p,u,\overline{\theta}))\}.$$

Again $h_{\theta_n}(\cdot) \to h_2(\cdot)$ uniformly on $\Delta_\epsilon \cup \Delta_\epsilon(1) \cup \Delta_\epsilon(2)$. Continuing the same process (assuming $\Delta_\epsilon(k_0) \neq \emptyset$), we have

$$\overline{\Gamma} + h_{k_0}(p) = \min_{u\in U}\{p'C + \sum_{y\in Y} V(y,p,u,\overline{\theta})h_{k_0}(T(y,p,u,\overline{\theta}))\} \tag{2.12}$$

for $p \in \Delta_\epsilon(k_0)$. Since $h_{\theta_n}(\cdot) \to h_{k_0}(\cdot)$ uniformly on $\Delta_\epsilon \cup \Delta_\epsilon(1) \cdots \cup \Delta_\epsilon(k_0) = \Delta$, (2.12) holds for all $p \in \Delta$, i.e., $(\overline{\Gamma}, h_{k_0}(\cdot))$ is a solution to the ACOE. Clearly $h_{k_0}(\cdot)$ is concave and $h_{k_0}(\overline{p}) = 0$. By the uniqueness, $\overline{\Gamma} = \Gamma_{\overline{\theta}}$ and $h_{k_0}(\cdot) = h_{\overline{\theta}}(\cdot)$ and so $h_\theta(p)$ is continuous in θ for all $p \in \Delta$. The rest of the proof is same as that of Theorem 2. **Q.E.D.**

References

[1] K.J. Åström, "Optimal control of Markov processes with incomplete state information," *J. Math. Anal. Appl.*, vol. 10, pp. 174-205, 1965.

[2] E. Fernández-Gaucherand, *Controlled Markov processes on the infinite planning horizon: Optimal and adaptive control*, Ph.D. dissertation, The Univ. of Texas at Austin, Aug., 1991.

[3] O. Hernández-Lerma, *Adaptive Markov control processes*, Springer-Verlag, N.Y., 1989.

[4] P.R. Kumar and P. Varaiya, *Stochastic systems: estimation, identification and adaptive control*, Prentice-Hall, Englewood Cliffs, N.J., 1986.

[5] R.T. Rockafellar, *Convex analysis*, Princeton University Press, Princeton, N.J., 1971.

REMARKS ON IDENTIFICATION OF CAUSAL DYNAMICAL SYSTEMS

Glenn K. Heitman
Department of Electrical Engineering
University of Akron
Akron, OH 44325-3904

Abstract

Preliminary results concerning the identification of classes of causal time-systems are presented. The identification is required to be accomplished in a finite length of time by means of input-output observations. This is done by converting the dynamic problem into a static one of identification of a class of mappings by a finite number of input-output tests for which results are known.

1. Introduction

This paper presents some observations and preliminary results concerning the identifiability of causal dynamical systems. The basic identification problem of concern here is this: from prior knowledge and observation of outputs produced by known inputs it is desired to construct a mathematical model for the unknown physical system that allows one to predict to within a specified tolerance the system output resulting from an arbitrarily chosen input belonging to a specified class. To be more precise, let there be given a class U of allowable inputs, a class Y of outputs, a class $\mathcal{G}$ of input-output mappings, $F : U \to Y$, to which the unknown system belongs, and another class $\mathcal{H}$ of identification mappings from U into Y. Then based on input-output experiments $y = F(u)$, we wish to choose an identification $\hat{F} \in \mathcal{H}$ for the unknown system $F \in \mathcal{G}$. (We shall describe in Section 2 the sense in which $\hat{F}$ approximates F.) The fundamental requirement that we impose is that the identification model $\hat{F}$ be determined on the basis of a *finite* number of input-output experiments, say

$$y_n = F(u_n),\ n = 1, 2, \ldots, N,$$

for some N.

The basic problem of identifiability based on a finite number of input-output tests is considered in detail in [2]; the main results are quoted in Theorems 1, 2, and 3 in Section 2. This problem may be called the "static" identification problem. In this paper we consider the "dynamic" identification problem—we shall translate the problem of identification of causal time-systems by means of a single test of finite time duration into the basic static problem. We shall consider both systems having bounded memory and unbounded memory.

2. Preliminaries

In this section we gather together results from [2] about identifiability that we shall need.

We begin by describing conditions under which a class $\mathcal{H}$ of systems is exactly identifiable. As mentioned in the Introduction, we consider identification based on a finite sequence of input-output tests

$$y_n = F(u_n),\ F \in \mathcal{H},\ n = 1, 2, \ldots, N.$$

Since we want to consider F as variable we rewrite the equation $y = F(u)$ as $y = u(F)$. Then the sequence of N input-output experiments is described by the equation

$$y^N = u^N(F),\ F \in \mathcal{H},$$

where $y^N = (y_1, \ldots, y_N)$ and where the mappping $u^N : \mathcal{H} \to Y^N$ is defined by

$$u^N(F) = (F(u_1), \ldots, F(u_N)).$$

Let $u^N_{\mathcal{H}}$ denote the restriction of u^N to its range $u^N(\mathcal{H})$. If $u^N_{\mathcal{H}}$ is invertible, then an exact identification of $F \in \mathcal{H}$ is given by

$$F = (u^N_{\mathcal{H}})^{-1}(y^N),\ y^N \in u^N(\mathcal{H}).$$

The following theorem describes an important case in which exact identification is possible. Note that the classes $\mathcal{G}$ (the unknown systems) and $\mathcal{H}$ (the identification models) are the same since we are considering exact identification.

Theorem 1. (Theorem 2 of [2]) *For any input space U and any finite-dimensional class $\mathcal{H}$ of systems, there exists a test set $(u_1, \ldots, u_N)$ in U such that every $F \in \mathcal{H}$ is exactly identifiable by the test set. N need not be larger than the dimension of $\mathcal{H}$.*

The following very simple example illustrates Theorem 1. Let the input space be $U = [0, 1]$ and the output space be $Y = \mathbb{R}$. Let the class $\mathcal{H}$ of systems be the space of all homogeneous polynomials of degree 2 or less on U; $\mathcal{H}$ is a linear space of dimension 2. Let u_1, u_2 be any two distinct points in U, and for $F \in \mathcal{H}$ let $y_1 = F(u_1)$ and $y_2 = F(u_2)$ be the corresponding test outputs. It is easy to show that $\hat{F} \in \mathcal{H}$ defined by

$$\hat{F}(u) = \frac{u_2^2 y_1 - u_1^2 y_2}{u_1 u_2^2 - u_1^2 u_2} u + \frac{u_1 y_2 - u_2 y_1}{u_1 u_2^2 - u_1^2 u_2} u^2$$

provides an exact identification for F; that is, we can exactly identify any $F \in \mathcal{H}$ by using the test set (u_1, u_2).

Definition 1. *If an identification model is given by*

$$\hat{F}(u) = \sum_{n=1}^{N} \gamma_n(u) F(u_n) \tag{1}$$

for some finite integer N where each $\gamma_n(u) = \gamma_n(u; u_1, \ldots, u_N)$, $n = 1, \ldots, N$, is a scalar function of u satisfying

(i) $\gamma_n(u_i) = \delta_{ni}$

(δ_{ni} is the Kronecker delta) then the identification is said to be a weak interpolation. If in addition each γ_n satisfies

(ii) $0 \leq \gamma_n(u) \leq 1$, $u \in U$

(iii) $\sum_{n=1}^{N} \gamma_n(u) = 1$, $u \in U$

then the identification is an interpolation. The γ_n *are called interpolation functionals with respect to the test set.*

If in Theorem 1 the output space Y is also finite-dimensional then the identification is an interpolation [2].

We shall now summarize results concerning the more important problem of approximate identifiability of a class of systems. We assume that U is a metric space with distance d and Y is a Banach space with norm $\|\cdot\|$. Then Y^N is a Banach space when equipped with the norm

$$\|y^N\| = \max_{i=1,\ldots,N} \|y_i\|.$$

$\mathcal{C}^\infty(U,Y)$ is the Banach space of continuous mappings from U into Y with bounded range and norm $\|F\| = \sup_{u \in U} \|F(u)\|$, and $\mathcal{B}^\infty(U,Y)$ is the Banach space of mappings from U into Y with bounded range and the same norm. Any space of mappings denoted by $\mathcal{H}$ or $\mathcal{G}$ is then at least a metric subspace of $\mathcal{C}^\infty$ or $\mathcal{B}^\infty$. (Hence we are considering identifications $\hat{F}$ which are uniform approximations of F.) We first need to impose more conditions on the interpolation functionals of Definition 1.

Definition 2. *Let* $\{\gamma_1, \ldots, \gamma_N\}$ *be a set of interpolation functionals with respect to the test set* $(u_1, \ldots, u_N)$*; i.e. they are real-valued functions on* U *satisfying conditions (i),(ii),(iii) of Definition 1. They are called* continuous *interpolation functionals if*

(iv) each γ_n *is continuous on* U*,*

and, for $\delta > 0$ *they are called* δ-local *interpolation functionals if*

(v) $\gamma_n(u) = 0$ *whenever* $d(u, u_n) \geq \delta$, $n = 1, \ldots, N$.

The two main results about approximate identifiability are quoted here as Theorems 2 and 3.

Theorem 2. (Theorem 3 of [2]) *Let* Y *be finite-dimensional. Let* U *be a compact metric space and let the class* $\mathcal{G}$ *of unknown systems be a compact metric subspace of* $\mathcal{C}^\infty(U,Y)$*. For a given* $\varepsilon > 0$ *there exists a test set* $(u_1, \ldots, u_N)$ *and a set of continuous local interpolation functionals* $\{\gamma_1, \ldots, \gamma_N\}$ *such that* $\hat{F} \in \mathcal{C}^\infty$ *defined by*

$$\hat{F}(u) = \sum_{n=1}^{N} \gamma_n(u; u_1, \ldots, u_N) F(u_n) \tag{2}$$

satisfies $\|F - \hat{F}\| \leq \varepsilon$ *for all* $F \in \mathcal{G}$.

In other words, under the stated hypotheses on U, Y, and $\mathcal{G}$, there is a continuous local interpolative ε-identification for the class $\mathcal{G}$. In the following theorem U is not necessarily compact and Y is not necessarily finite-dimensional, and the conclusion of Theorem 2 still holds except that the interpolation is no longer local.

Theorem 3. (Theorem 4 of [2]) *Let U be a metric space and Y a Banach space. Let $\mathcal{G}$ be totally bounded in $\mathcal{C}^\infty(U,Y)$ and let*

$$\mathcal{G}(U) \stackrel{\text{def}}{=} \{F(u) : F \in \mathcal{G},\, u \in U\}$$

be totally bounded in Y. For a given $\varepsilon > 0$ there exists a test set $(u_1, \ldots, u_N)$ and a set of continuous interpolation functionals $\{\gamma_1, \ldots, \gamma_N\}$ such that $\hat{F} \in \mathcal{C}^\infty$ defined by

$$\hat{F}(u) = \sum_{n=1}^{N} \gamma_n(u; u_1, \ldots, u_N) F(u_n) \tag{3}$$

satisfies $\|F - \hat{F}\| \leq \varepsilon$ for all $F \in \mathcal{G}$. If Y is finite-dimensional then the identification model $\hat{F}$ is finitely parametrized.

The following example illustrates Theorem 2. Consider linear lumped-parameter system models:

$$\dot{x}(t) = Ax(t) + Bu(t), \quad x(0) = 0, \tag{4}$$

$$y(t) = Cx(t), \tag{5}$$

where $x(t) \in \mathbb{R}^n$ and $u(t) \in \mathbb{R}$ for $t \in [0,T]$ (T fixed), and $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times 1}$, and $C \in \mathbb{R}^{1\times n}$ are unknown constant matrices. The identification problem then is to determine the matrices A, B, and C from input-output experiments. For any system modeled by Equations (4) and (5) we shall assume that

1. the system is controllable and observable,
2. the matrix A is in Jordan form, and
3. the eigenvalues $\lambda_1, \ldots, \lambda_n$ of A are distinct, non-zero, and in the left-half plane.

These assumptions imply that no entry of B is zero, no entry of C is zero, and that A is a diagonal matrix with the eigenvalues on the diagonal. Furthermore, we can assume that $C = [1\,1\,\cdots\,1]$ because any system described by Equations (4) and (5) is equivalent to one in which C consists of all 1's. (See for example [1].) If we denote the entries of the $n \times 1$ matrix B by $b_1, \ldots, b_n$, then the identification problem is to (approximately) determine the parameters $\lambda_1, \ldots, \lambda_n$ and $b_1, \ldots, b_n$.

The input-output representation of a system modeled by Equations (4) and (5) is given by the standard formula

$$y(t) = [F(u)](t) = C\int_0^t e^{A\tau} Bu(t-\tau)d\tau = \int_0^t \Big(\sum_{i=1}^{n} b_i e^{\lambda_i \tau}\Big) u(t-\tau)d\tau, \tag{6}$$

for $t \in [0,T]$. Equation (6) defines a mapping $F : U \to Y$; then for given classes $\mathcal{A} \subset \mathbb{R}^{n\times n}$ and $\mathcal{B} \subset \mathbb{R}^{n\times 1}$ of matrices A and B we have a class $\mathcal{G}$ of such input-output mappings. In order to apply Theorem 2, let U^0 be a finite-dimensional subspace of $L_2[0,T]$. Each F defined by (6) is linear, so each $F(U^0)$ is a finite-dimensional subspace of $L_2[0,T]$, having the same dimension as U^0. Hence, we may take the output space Y to be a finite-dimensional subspace of $L_2[0,T]$. We take the input space U to be a compact subset of U^0 (say the closed unit ball). Theorem 2 also requires that $\mathcal{G}$ be compact in $\mathcal{C}^\infty(U,Y)$. To obtain a compact $\mathcal{G}$, we proceed as follows. In the linear space $\mathbb{R}^{n\times n}$ of $n \times n$

matrices let the norm be the usual linear operator norm: $\|A\| = \sup_{\|x\|=1} \|Ax\|$, where the norm $\|x\|$ in $\mathbb{R}^n$ is the Euclidean norm. (For diagonal A we have $\|A\| = \lambda_{\max}^{1/2}$.) The norm $\|B\|$ in $\mathbb{R}^{n\times 1}$ is just the Euclidean norm. Now suppose that the class $\mathcal{A} \subset \mathbb{R}^{n\times n}$ is a closed ball of radius r_A about a nominal A_0; likewise, $\mathcal{B} \subset \mathbb{R}^{n\times 1}$ is a closed ball of radius r_B about a B_0. That is, we assume nominal A_0 and B_0 in the model specified by Equations (4) and (5), and we represent our modeling error, or parameter uncertainty, by the balls $\mathcal{A}$ and $\mathcal{B}$. It is then straightforward to show that the resulting class $\mathcal{G}$ of input-output systems is a compact subset of $\mathcal{C}^\infty(U, Y)$. Hence, Theorem 2 applies: there is a finite test set such that every $F \in \mathcal{G}$ can be approximately identified by an interpolation of the form of Equation (1). We shall see in the next section that we can in fact exactly identify this class, even though Theorem 1 does not apply.

3. Classification of Systems

We now consider identification of causal time-systems. First, we shall classify classes of causal time-invariant systems into three types, the type depending upon whether a system in the class can be exactly identified in an arbitrarily short time, in some finite time, or not at all in any length of time. We assume in each case that the zero function belongs to the input space U and that each system F carries 0 into 0.

Type I. The class of unknown systems is such that for any $\delta > 0$ an exact identification is possible for some experiment performed on the time interval $[0, \delta]$.

Example. Consider again the class consisting of n-dimensional single-input single-output time-invariant linear systems described by Equations (4) and (5), under the same assumptions.

The input-output description is again given by the formula

$$y(t) = C\int_0^t e^{A\tau}Bu(t-\tau)d\tau = C\int_0^t e^{A(t-\tau)}Bu(\tau)d\tau = \int_0^t \Big(\sum_{i=1}^n b_i e^{\lambda_i(t-\tau)}\Big)u(\tau)d\tau, \qquad (7)$$

for $t \in [0, T]$.

Suppose that the input $u(t)$ is the unit step function $s(t)$:

$$s(t) = \begin{cases} 1, & \text{for } t \geq 0 \\ 0, & \text{for } t < 0. \end{cases}$$

Let y denote the corresponding output calculated from (7):

$$y(t) = \frac{b_1}{\lambda_1}(e^{\lambda_1 t} - 1) + \cdots + \frac{b_n}{\lambda_n}(e^{\lambda_n t} - 1) = d_1(e^{\lambda_1 t} - 1) + \cdots + d_n(e^{\lambda_n t} - 1),$$

for $t \in [0, T]$, where $d_i \stackrel{\text{def}}{=} b_i/\lambda_i$ for $i = 1, 2, \ldots, n$.

We wish to show that the λ_i's and the b_i's can be determined uniquely by observing $y(t)$ over any arbitrarily short interval $[0, \delta]$. Suppose that there are two sets $\{\lambda_1, \ldots, \lambda_n\}$, $\{b_1, \ldots, b_n\}$ and $\{\lambda'_1, \ldots, \lambda'_n\}$, $\{b'_1, \ldots, b'_n\}$ such that

$$d_1(e^{\lambda_1 t} - 1) + \cdots + d_n(e^{\lambda_n t} - 1) = d'_1(e^{\lambda'_1 t} - 1) + \cdots + d'_n(e^{\lambda'_n t} - 1)$$

where of course $d'_i = b'_i/\lambda'_i$. Then

$$d_1 e^{\lambda_1 t} + \cdots + d_n e^{\lambda_n t} - d'_1 e^{\lambda'_1 t} - \cdots - d'_n e^{\lambda'_n t} = \text{const} \qquad (8)$$

for $t \in [0, T]$. Suppose for example that $\lambda_1' \neq \lambda_1$. Differentiate with respect to t in Equation (8):

$$\lambda_1 d_1 e^{\lambda_1 t} + \cdots + \lambda_n d_n e^{\lambda_n t} - \lambda_1' d_1' e^{\lambda_1' t} - \cdots - \lambda_n' d_n' e^{\lambda_n' t} = 0$$

or

$$b_1 e^{\lambda_1 t} - b_1' e^{\lambda_1' t} + \cdots + b_n e^{\lambda_n t} - b_n' e^{\lambda_n' t} = 0.$$

The first two terms on the left are unequal and not zero, so regardless of whether $\lambda_k = \lambda_k'$ or $\lambda_k \neq \lambda_k'$ for $k = 2, 3, \ldots, n$, this is a trivial linear combination of a set of independent functions—a contradiction. Hence, we must have $\lambda_i = \lambda_i'$ for all i. Likewise, we must also have $b_i = b_i'$ for all i.

Hence, under the stated assumptions, the dynamical system of Equations (4) and (5) can be *exactly* identified by observing the input and output functions over any arbitrarily short interval.

Type II. The class of unknown systems is such that for some $T_0 \in (0, \infty)$ an exact identification is possible for some experiment (or set of experiments) of time duration T_0 (or of course of duration $T > T_0$).

Example. Let the class $\mathcal{H}$ of unknown systems consist of time-invariant systems F each having memory $m \in (0, M)$ and such that the "truncated systems" formed from the systems in $\mathcal{H}$ yield a class that is exactly identifiable. (Truncated systems will be precisely defined subsequently; here we need only remark that they take the input to the system F over the intervals $[0, m+b]$, $[m+b, 2(m+b)]$, ... and the corresponding outputs are observed over $[m, m+b]$, $[2m+b, 2(m+b)]$, ...) In particular, if $\mathcal{H}$ is a linear space of finite dimension, this holds. If N experiments are required to obtain a determination function then T_0 can be taken to be $N(m+b)$. Of course it may be that a smaller T_0 will do.

Remarks. (a) Any system of Type I is of course also of Type II. Nevertheless it is useful to distinguish the two types.

(b) The example given for Type I does not involve bounded memory, but it also an example of a Type II class. In general, bounded memory is not requisite for Type II, but one must be able to infer in some way the effects of unbounded memory from experiments performed during a finite time. On the other hand, bounded memory does not by itself imply that a class is of Type II.

Type III. The class of unknown systems is such that no experiment (or set of experiments) performed on a finite time interval can result in an exact identification.

Example 1. The class $\mathcal{H}$ of unknown systems consists of all finite duration delays. This class is parametrized by one parameter, namely the delay. If one were willing to allow experiments that required the infinite positive time-axis for their performance, then this class would be exactly identifiable with one experiment. Hence Type III includes classes of systems that could be considered exactly identifiable in an abstract sense.

Example 2. This is an example of a class of systems that cannot be exactly identified in any finite time because the class is too large. The class $\mathcal{H}$ is known to be *one* of $\mathcal{H}_1$, $\mathcal{H}_2$, ..., where $\mathcal{H}_i \subset \mathcal{H}_{i+1}$ and the time required for identification of $\mathcal{H}_i$ is T_i, with $T_i \uparrow \infty$. The class is not identifiable because after a finite number of tests there is *no way of knowing* whether or not the system has been identified. This kind of example is clearly related to the question of *verification* of identification.

4. Bounded Memory Systems

In this section we shall describe how the problem of identification of time-invariant causal systems having bounded memory by means of input-output data taken over a finite interval of time can be translated into the problem of identification by means of a finite sequence of experiments, as described in Section 2. In the discussion to follow, all input and ouput spaces are restricted to be L_2 spaces.

We begin by describing appropriate input and output spaces. All time functions (inputs or outputs) are to be real-valued functions of a real variable, or they are to be vector-valued functions with a finite number of real-valued components. The usual operator of projection on the past, P_t, is defined for all t by

$$(P_t f)(s) = \begin{cases} f(s), & s \leq t \\ 0, & s > t. \end{cases}$$

The operator of left-shift, L_t, is defined for all t by

$$(L_t f)(s) = f(s+t).$$

The notation $R_t = L_{-t}$ will be used for the right-shift operator. The input space U is to be a shift-invariant subset of the space of locally square-integrable functions of time. That is, U must satisfy $L_t U = U$ for all t, and $\|(P_{t+T} - P_t)u\|_{L_2} \leq K$ for all t and T and some $K = K(T,u)$. U is also required to satisfy one additional condition: the restriction $U|I$ to any finite interval I is to be a totally bounded subset of $L_2(I)$. The output space Y is to be the (shift-invariant) space of locally square-integrable functions of time. (If the input and output functions are vector-valued, they may take values in spaces of different dimension; say $\mathbb{R}^\mu$ for inputs and $\mathbb{R}^\nu$ for ouputs.) Any class $\mathcal{G}$ or $\mathcal{H}$ of unknown systems is required to be a subset of $\mathcal{C}^\infty(U,Y)$; the mappings are to be causal, time-invariant, and are to have finite memory $m \geq 0$. The causality requirement is described by the condition

$$P_t F = P_t F P_t, \text{ for all } t.$$

The requirement that each F have finite memory m means that

$$(I - P_t)F = (I - P_t)F(I - P_{t-m}), \text{ for all } t.$$

In the discussion of identification to follow, it is not the time systems (Y, F, U) that play a direct role; rather we obtain a *sequence* of input-output tests by means of the "truncated systems", which we shall now describe. To begin, define for any $b > 0$ the spaces

$$\begin{aligned} U_{t-b-m,t} &= (P_t - P_{t-b-m})U, \text{ for all } t, \\ Y_{t-b,t} &= (P_t - P_{t-b})Y, \text{ for all } t. \end{aligned}$$

It follows from the conditions imposed on U that $U_{t-b-m,t}$ is a totally bounded subset of $L_2[t-b-m,t]$ for each t and that $Y_{t-b,t} = L_2[t-b,t]$ for each t. By the shift-invariance of U and Y, each $U_{t-b-m,t}$ is isometrically isomorphic to $U_{-b-m,0}$ and each $Y_{t-b,t}$ is isometrically isomorphic to $Y_{-b,0}$. Now any causal $F : U \to Y$ with memory not exceeding m induces a mapping $\tilde{F}_t : U_{t-b-m,t} \to Y_{t-b,t}$. The *truncated system* corresponding to F is the map $F_t : U_{-b-m,0} \to Y_{-b,0}$ defined by

$$F_t(u) = (L_t \tilde{F}_t R_t)(u), \; u \in U_{-b-m,0}.$$

Because F is time-invariant, $F_t = F_0$, for all t. Rather than deal with the system $F : U \to Y$ we shall deal with the system $F_0 : U_{-b-m,0} \to Y_{-b,0}$.

Remark. We use the spaces $U_{-b-m,0}$ and $Y_{-b,0}$ because instantaneous measurements of L_2 functions are not defined; b represents a "measurement interval".

For any time t_1, given the input $u_{t_1-b-m,t_1} \in U_{t_1-b-m,t_1}$, observing

$$y_{t_1-b,t_1} = \tilde{F}_{t_1}(u_{t_1-b-m,t_1}) \in Y_{t_1-b,t_1}$$

constitutes one experiment. Put

$$\begin{aligned} u_1 &= L_{t_1}(u_{t_1-b-m,t_1}) \in U_{-b-m,0}, \\ y_1 &= L_{t_1}(y_{t_1-b,t_1}) \in Y_{-b,0}. \end{aligned}$$

Then the experiment can be written

$$y_1 = F_0(u_1).$$

Repeating this at times $t_2, t_3, \ldots$ results in a sequence of experiments

$$y_i = F_0(u_i),\ i = 1, 2, \ldots,$$

where $u_i = L_{t_i}(u_{t_i-b-m,t_i}) \in U_{-b-m,0}$ and $y_i = L_{t_i}(y_{t_i-b,t_i}) \in Y_{-b,0}$. If we stop at $i = N$ then of course we have obtained a finite sequence of experiments with test set $(u_1, \ldots, u_N)$ and the results of Section 2 can be applied.

5. Unbounded Memory Systems

The procedure described in Section 4 for obtaining a sequence of input-output experiments cannot be used if the systems $F \in \mathcal{G}$ have infinite memory. If, however, the unbounded memory systems can be approximated by systems having uniformly bounded memory, then the procedure described in Section 4 can be used; one hopes that only a small approximation error is introduced.

One approach to the approximation of unbounded memory systems by bounded memory systems is suggested by the concept of *tapered* input spaces, introduced by Root; see [5]. To begin, we again require that U be a shift-invariant subset of locally square-integrable functions, that $U|I$ be compact in $L_2(I)$ for all intervals I, and that Y is the shift-invariant space of locally square-integrable functions. In the space $\mathcal{C}^\infty(U, Y)$ let $\mathcal{C}_c^\infty$ denote the linear subspace of causal mappings, and let $\mathcal{C}_{cm}^\infty$ denote the linear subspace of causal mappings having finite memory m. The class $\mathcal{G}$ of unknown systems is assumed to be a subset of $\mathcal{C}_c^\infty$, but not necessarily a subset of $\mathcal{C}_{cm}^\infty$. For any $F \in \mathcal{C}^\infty$ define the *causal part* of F to be the mapping F_c defined by

$$F_c = FP_t, \quad \text{for all } t.$$

In general $F \in \mathcal{C}^\infty$ does not imply that $F_c \in \mathcal{C}_c^\infty$; i.e., the causal part of F need not be continuous simply because F is. Also for any $F \in \mathcal{C}^\infty$ define the *causal memory m part* of F to be the mapping F_{cm} defined by

$$F_{cm} = F(P_t - P_{t-m}), \quad \text{for all } t.$$

Again, $F \in \mathcal{C}^\infty$ does not imply $F_{cm} \in \mathcal{C}_{cm}^\infty$. Assume, however, that $F \in \mathcal{C}^\infty$ and $F_{cm} \in \mathcal{C}_{cm}^\infty$. Under what condition is F_{cm} close to F in the $\mathcal{C}^\infty$ topology? Suppose that there is

an $F' \in \mathcal{C}_{cm}^{\infty}$ with $\|F' - F\| \leq \varepsilon$ for some $\varepsilon > 0$. Then for any $\delta > 0$ there is $u \in U$ and a t_0 such that

$$\begin{aligned} \|F' - F_{cm}\|_{\infty} &\leq \|F'(u) - F(P_{t_0} - P_{t_0-m})(u)\|_Y + \delta/2 \\ &= \|F'(P_{t_0} - P_{t_0-m})(u) - F(P_{t_0} - P_{t_0-m})(u)\| + \delta/2 \\ &\leq \|F' - F\| + \delta/2 \\ &\leq \varepsilon + \delta/2. \end{aligned}$$

Hence,

$$\|F - F_{cm}\| \leq \|F - F'\| + \|F' - F_{cm}\| \leq 2\varepsilon + \delta/2.$$

That is, if there is a causal memory m mapping F' close to F, then the causal memory m part of F is also close to F. See also [4].

In order to apply the procedure used in Section 4, we need to impose a further condition on the input space. The idea is this: if the remote past of each $u \in U$ does not contribute much weight to $\|u\|$, then the approximation of a system $F \in \mathcal{C}_c^{\infty}$ by its causal memory m part, F_{cm}, should be good. Then, proceeding as in Section 4, we obtain a sequence of input-output experiments based on the truncated systems $(F_{cm})_t$. The mathematical statement of the condition that the remote past of each input does not contribute much weight to the norm is that for all $\varepsilon > 0$, $c > 0$, and t there is a $\delta > 0$ such that

$$\|P_t u\| \leq \|(P_t - P_{t-\delta})u\| + \varepsilon,$$

provided $\|P_\tau u\| \leq c$ for all $\tau \leq t$. (The norm is the L_2 norm.) An input space satisfying this condition is said to be *tapered*. (See [5] for a general development of the theory of tapered spaces.) If the input space is tapered then replacing F by F_{cm} introduces a small approximation error, and we can proceed as in Section 4.

6. Conclusion

We have discussed some preliminary results concerning the identification of causal dynamical systems from a finite duration input-output experiment. This is achieved for finite memory systems by dividing the inputs into segments of duration $b + m$ and considering the induced truncated systems F_t. The dynamic problem is thus reduced to the static problem described in Section 2. For infinite memory systems we still divide the inputs into finite duration segments, but the truncated systems are approximations; if the input space is tapered, then the approximation error is small.

A related question that we have not addressed in detail because of space limitations is the complexity of the identification. This is considered for the static problem in [3]. In the static problem complexity is, roughly, the number of input-output experiments that must be performed to achieve an identification. In the dynamic problem this notion of complexity becomes the time duration required of an experiment to achieve an identification. The translation from the static results in [3] to the dynamic problem is not, however, immediate. Suppose that the class $\mathcal{G}$ of truncated systems is such that either Theorem 1 or Theorem 2 applies; then there is a finite test set $(u_1, \ldots, u_N)$ in $U_{-b-m,0}$ that will either exactly or approximately identify $\mathcal{G}$. Unfortunately, there is nothing to tell us how to choose the u_i's—only N of them are needed, but these N inputs may correspond to widely separated segments of $u(t)$, and so the time required to achieve an identification may be much more than $N(b + m)$. Obviously, we must impose more

conditions on the inputs. One simple condition we could require is that the inputs should not change too much over short time intervals; i.e., for all $\eta > 0$ there is a $\xi > 0$ such that $\|L_\xi u - u\| \leq \eta$ for all $u \in U$. This may be a reasonable line of research to pursue.

We shall conclude with some questions that are suggested by our discussion. The first is *how* the results of Section 2 should be applied to the dynamic problem. There are two protocols for performing the identification in the dynamic problem: one is to fix the input space $U_{-b-m,0}$, which must be big enough to contain all "typical" inputs, and the other is to perform the experiment on-line, in which case the input space $U_{-b-m,0}$ is generated as the input runs, and now the input must run long enough so that the input space contains all "typical" inputs. Second, the meaning of "typical" as used above must be properly dealt with. And third, upper and lower error bounds for approximating the infinite memory system by a bounded memory system with tapered input space are needed.

References

[1] Chen, Chi-Tsong, *Linear System Theory and Design*, Holt, Rinehart, and Winston, New York, 1984.

[2] Heitman, G.K., "Identifiability of classes of input-output systems", *J. Math. Analysis and Applications*, to appear.

[3] Heitman, G.K., "Complexity of identification of input-output systems", *J. Math. Analysis and Applications*, to appear.

[4] Root, W.L., "On the modeling of systems for identification. Part II: Time-varying systems", *SIAM J. Control*, v.13, No.4, pp.945-974 (1975).

[5] Root, W.L., "Considerations regarding input and output spaces for time-varying dynamical systems", *Appl. Math. Optim.*, **4**, pp.365-384 (1978).

Universal Upper Bounds on the Packet Loss Probability of Statistical Multiplexing in High-Speed Networks[1]

Emmanouel (Manos) Varvarigos
University of California - Santa Barbara
Department of Electrical and Computer Engineering
Santa Barbara, CA 93106
E-mail: manos@ece.ucsb.edu

Abstract

The performance of statistical multiplexing in high-speed networks is analyzed for the case where N independent sessions share a link. We derive a general upper bound on the packet loss probability on the link without having to specify a particular model for the traffic behavior of individual sessions. This is helpful in the usual case where the session statistics are either complex or unknown. When the sessions sharing the link follow paths that intersect prior to that link, the sessions become correlated. In that case we modify our model to obtain an upper bound that takes into account the dependencies created. We also provide an upper bound on the packet loss probability when a leaky-bucket scheme is used to shape the input traffic.

1. INTRODUCTION

The main advantage of packet switching over other switching formats is the improvement in network utilization through the process of statistical multiplexing. In contrast to circuit switching, where the capacity allocated to a session may remain idle even if other sessions want to use it, in packet switching a link is used whenever there is a packet that wants to cross it. Packet switching has, however, a number of drawbacks, especially when the buffer space per node is limited and packets must occasionally be dropped.

A solution that has been proposed to avoid the dropping of packets is deflection routing (see, for example, [4], [5], and [2]), where packets are misrouted instead of dropped. Deflection routing has the disadvantage that packets do not always follow shortest paths to their destination, and they may arrive out of order. This is undesirable for high speed networks where a large number of packets may have to be resequenced. Another scheme that avoids the dropping of packets is the Conflict Sense Routing protocol proposed in [6]. This protocol is efficient in terms of link utilization, but it requires a pretransmission delay.

For connections whose rate remains constant with time, the combination of virtual circuit switching (which is a form of packet switching) with capacity reservation protocols (see [1]) and special queueing structures (see [3]) can provide loss-free communication. Reservation protocols are, however, inappropriate for sessions with very strict delay requirements, because of the pre-transmission delay associated with the reservations. Loss-free transmission can still be achieved for delay-sensitive traffic through buffering and backpressure protocols without using reservations, provided that the rate changes only slowly with time.

Loss-free communication is, however, difficult to achieve for bursty, delay-sensitive sessions. When the delay requirements are strict, it may be undesirable to smoothen bursty traffic through the use of input control, because of the delay introduced at the input. Back-pressure type of protocols are also inappropriate for bursty sessions in high-speed networks, because the large (relative to packet transmission times) propagation delays make them slow to react to rapid

[1] Research supported under contracts by ARPA under Contract DABT63-93-C-0039, and by NSF under Grant NSF-RIA-08930554.

changes in the transmission rates. For applications where the previous options are not desirable, the loss of packets due to buffer overflow is unavoidable. Therefore, reliable estimates on the packet loss probability are an integral part of any policy for the admission control of bursty, delay-sensitive sessions. A new session should be rejected whenever the estimated packet loss probability guaranteed by the network is unacceptable for that session, or whenever its admission would endanger the grade of service already agreed between the network and the existing users.

In this paper we focus on a link $\mathcal{L}$ of the network on which N connections are statistically multiplexed. Each session is characterized by two parameters. The first parameter is a measure of the long-term average rate of the session, while the second is related to its maximum short-term average rate over periods of duration T. We initially ignore any correlations among the sessions, and we obtain an upper bound on the packet loss probability on link $\mathcal{L}$. This upper bound does not require any knowledge of the particular probabilistic characteristics of the sessions (other than the two parameters mentioned aboved), and it holds for any probability distribution of the session arrival processes.

When the sessions that are multiplexed on link $\mathcal{L}$ share links prior to their arrival at $\mathcal{L}$, the packet arrivals that correspond to different sessions become correlated. We assume that a special queueing structure (namely, the stop-and-go queueing proposed by Golestani [3]) is used at the nodes. We obtain an upper bound on the packet loss probability that takes into account the session interactions, and does not use any approximating assumptions.

We also consider the case where a leaky-bucket scheme is used to control the entry of packets into the network. We modify our results to provide upper bounds on the packet loss probability that depend only on the parameters of the leaky-bucket scheme, and are independent of the source probabilistic characteristics.

The organization of the paper is the following. In Section 2 we describe the communication model, and we introduce the notation used. In Sections 3 and 4 we give upper bounds on the overflow probability and the packet loss probability on a network link. In Section 5 we modify our results to take into account the correlation among the streams that are multiplexed on the link. Finally, in Section 6 we consider the case where a leaky-bucket scheme is used for input control.

2. COMMUNICATION MODEL AND NOTATION

We assume that packets have equal length, and they require one unit of time (or slot) for transmission over a link. Transmission rates are normalized to one, and they are measured in packets per slot (therefore, the capacity of a link is equal to one). We focus on a particular link $\mathcal{L}$ of the network on which N sessions are statistically multiplexed, and we let C, $0 \leq C \leq 1$, be the portion of the capacity that is allocated to these sessions (see Fig. 1). There is also buffer space for $B+1$ packets (excluding packets that are under transmission) that is allocated to these N sessions. The session durations, measured in slots, are assumed to be large enough, so that they can be approximated to be equal to infinity.

We denote by $A_i(t,T)$ the number of packets of session i that arrive at the buffer of link $\mathcal{L}$ in the time interval $(t-T,t]$. For a given T (measured in slots), we define the *T-sliding-window rate* $\lambda_i^T(t)$ of session i at time t as

$$\lambda_i^T(t) = \frac{A_i(t,T)}{T}, \quad i = 1,2,\ldots,N.$$

The size T of the sliding window used in the definition of $\lambda_i^T(t)$ will be treated as a parameter in our preliminary results, and it will be assigned specific values only when motivated by the analysis. At a given time slot t, $\lambda_i^T(t)$ is a random variable with a particular (but unknown) probability density function $p_{\lambda_i^T(t)}(x)$. The T-sliding-window rates $\lambda_i^T(t)$ are assumed to be independent (the case where they are correlated due to session interactions will be treated in Section 5) with known mean and maximum value. In particular, we let

$$R_i^T = \max_t \lambda_i^T(t)$$

be the *peak T-sliding-window rate* of source i, and

$$r_i = E[\lambda_i^T(t)]$$

(assumed to be independent of t and T) be the *average rate* of session i. We also assume that $\sum_{i=1}^{N} r_i \leq C$, so that the sum of the rates does not exceed the capacity allocated to the sessions.

We define the *smoothness coefficient* β_i^T of source i as the ratio

$$\beta_i^T = \frac{r_i}{R_i^T}.$$

Small values of the smoothness coefficient β_i^T correspond to bursty sessions, while $\beta_i^T = 1$ corresponds to a constant rate session. Note that R_i^T is monotonically decreasing with T, and it converges to r_i as T tends to infinity. It is also easy to see that $\beta_i^T \to 1$ as $T \to \infty$.

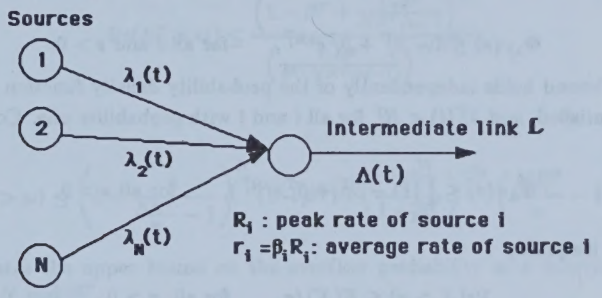


Figure 1: N independent sources sharing a communication link $\mathcal{L}$.

We let

$$\Lambda^T(t) = \sum_{i=1}^{N} \lambda_i^T(t)$$

be the total T-sliding-window rate of packet arrivals on link $\mathcal{L}$ at time slot t.

In the following sections we will obtain upper bounds on the packet loss probability on link $\mathcal{L}$ that depend only on the parameters r_i and R_i^T of the sessions, and assume no other knowledge about the rates $\lambda_i^T(t)$. They will hold uniformly for all choices of the probability density functions $p_{\lambda_i^T(t)}(x)$.

3. UPPER BOUND ON THE OVERFLOW PROBABILITY

For a given scalar ω we define the link overflow probability $\Pr(\Lambda^T(t) \geq \omega)$ as the probability that the time average of the total number of arrivals during the interval $(t-T, t]$ exceeds ω. In what follows we derive an upper bound on the link overflow probability; this bound will be used in Sections 4 and 5 to obtain upper bounds on the packet loss probability.

For a fixed t, $\lambda_i^T(t)$ is a random variable with known mean r_i, maximum value R_i^T, and (unknown) probability density function $p_{\lambda_i^T(t)}(x)$. For the purpose of the analysis of this and the next section, we will assume that the random variables $\lambda_i^T(t)$, $i = 1, 2, \ldots, N$, are independent. (In Section 5 we will modify the model to take into account possible correlations among the sessions.) Since we will be focusing on a particular time slot t, the dependence on t will be suppressed in order to simplify the notation when no confusion can arise.

We let $\Phi_{\Lambda^T}(s)$ be the characteristic function of $\Lambda^T(t)$, and $\Phi_{\lambda_i^T}(s)$ be the characteristic function of $\lambda_i^T(t)$. The independence of the $\lambda_i^T(t)$, $i = 1, 2, \ldots, N$, implies

$$\Phi_{\Lambda^T}(s) = E(e^{s\Lambda^T}) = \prod_{i=1}^{N} \Phi_{\lambda_i^T}(s). \tag{1}$$

From the definition of the characteristic function we have

$$\Phi_{\lambda_i^T}(s) = E(e^{s\lambda_i^T}) = \int_0^{R_i^T} e^{sx} p_{\lambda_i^T}(x)\, \mathrm{dx}. \tag{2}$$

Equation (2) together with the monotonicity of the exponential function and the relation

$$E(\lambda_i^T) = \int_0^{R_i^T} x p_{\lambda_i^T}(x)\, \mathrm{dx} = r_i = \beta_i^T R_i^T \tag{3}$$

imply

$$\Phi_{\lambda_i^T}(s) \leq 1 - \beta_i^T + \beta_i^T e^{sR_i^T}, \qquad \text{for all } i \text{ and } s > 0. \tag{4}$$

The preceding bound holds independently of the probability density function $p_{\lambda_i^T}(x)$, provided that Eq. (2) is satisfied, and $\lambda_i^T(t) < R_i^T$ for all i and t with probability one. Combining Eqs. (1) and (4) we get

$$\Phi_{\Lambda^T}(s) \leq \prod_{i=1}^{N} (1 - \beta_i^T + \beta_i^T e^{sR_i^T}), \qquad \text{for all } s > 0. \tag{5}$$

Using Markov's inequality

$$\Pr(X > a) \leq E(X)/a, \qquad \text{for all } a > 0,$$

with $X = e^{s\Lambda^T}$ and $s > 0$, we obtain the following inequality (which is equivalent to the Chernoff bound):

$$\Pr(e^{s\Lambda^T} > a) \leq \frac{\Phi_{\Lambda^T}(s)}{a}, \qquad \text{for all } a > 0,\ s > 0. \tag{6}$$

Equations (5) and (6) with $a = e^{s\omega}$ give

$$\Pr(\Lambda^T > \omega) \leq \frac{\prod_{i=1}^{N} (1 - \beta_i^T + \beta_i e^{sR_i^T})}{e^{s\omega}}, \qquad \text{for all } s > 0.$$

Letting $e^{s\omega} = k$, the preceding inequality can be rewritten as

$$\Pr(\Lambda^T > \omega) \leq \frac{1}{k} \prod_{i=1}^{N} (1 - \beta_i^T + \beta_i^T k^{\frac{R_i^T}{\omega}}), \qquad \text{for all } k > 1,$$

which gives

$$\Pr(\Lambda^T > \omega) \leq \min_{k>1} \frac{1}{k} \prod_{i=1}^{N} (1 - \beta_i^T + \beta_i^T k^{\frac{R_i^T}{\omega}}). \tag{7}$$

The expression at the right hand side of Eq. (7) is minimized for k that satisfies

$$\sum_{i=1}^{N} \frac{R_i^T \beta_i^T k^{\frac{R_i^T}{\omega}}}{1 - \beta_i^T + \beta_i^T k^{\frac{R_i^T}{\omega}}} = \omega.$$

For any given values of the parameters b_i^T and R_i^T, the preceding equation can be solved numerically with respect to k in order to obtain the most strict of the bounds of Eq. (7). Alternatively,

one can substitute different values for k in the right hand side of Eq. (7), and obtain several upper bounds on the overflow probability, from which the stricter can be chosen.

To carry the calculations further, we assume that for all sessions i we have $R_i^T \leq R^T$ and $r_i \leq r$, and we let $\beta^T = r/R^T$. Then Eq. (7) gives

$$\Pr(\Lambda^T > \omega) \leq \min_{k>1} \frac{1}{k} \left(1 - \beta^T + \beta^T k^{\frac{R^T}{\omega}}\right)^N . \tag{8}$$

The right hand side of Eq. (8) is minimized for

$$k = \left(\frac{1-\beta^T}{\beta^T(NR^T/\omega - 1)}\right)^{\omega/R^T} .$$

Substituting this value of k into Eq. (8), we obtain

$$\Pr(\Lambda^T > \omega) \leq \frac{\left(1 - \beta^T + \frac{1-\beta^T}{NR^T/\omega - 1}\right)^N}{\left(\frac{1-\beta^T}{\beta^T(NR^T/\omega - 1)}\right)^{\omega/R^T}},$$

or equivalently

$$\Pr(\Lambda^T > \omega) \leq \left(1 + \frac{1}{\frac{NR^T}{\omega} - 1}\right)^N (1-\beta^T)^N \left(\frac{\beta^T}{1-\beta^T}\right)^{\frac{\omega}{R^T}} \left(\frac{NR^T}{\omega} - 1\right)^{\frac{\omega}{R^T}} . \tag{9}$$

Figure 2 illustrates the upper bound on the overflow probability as a function of ω/R^T, for several values of N and β^T.

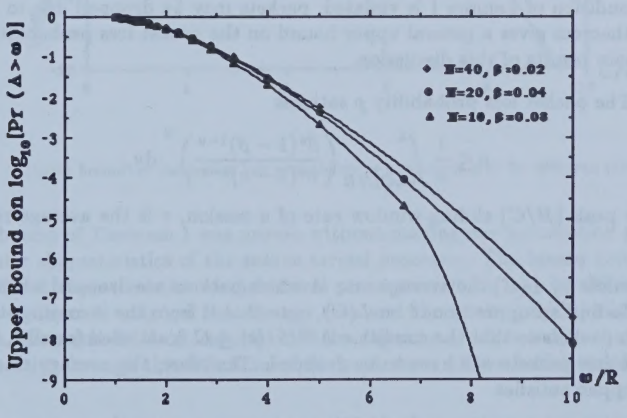


Figure 2: Upper bound on the overflow probability as a function of ω/R^T, for different values of N and β^T.

4. UPPER BOUND ON THE PACKET LOSS PROBABILITY

In this section we obtain an upper bound on the probability that a packet is dropped due to buffer overflow at link $\mathcal{L}$. To do so, we will use the bound on the overflow probability obtained in Section 3, with the parameter T taking appropriate values.

The following lemma, which gives a sufficient condition for loss-free communication on link $\mathcal{L}$, will be useful in our analysis.

Lemma 1: Let $T = \lfloor B/C \rfloor$, where $B+1$ is the size of the buffer of the link measured in packets (excluding the packets under transmission), and C is the capacity of the link allocated to the sessions. If $\Lambda^T(t) \leq C$ for all t, then no packets are lost due to buffer overflow.

Proof: Let $\mathcal{B}(t)$ be the number of packets in the buffer at time t (we use the convention that $\mathcal{B}(t)$ does not include the packet under transmission or the packets arriving during slot t; also packets arriving during a slot cannot be transmitted during the same slot). We assume that the buffer is initially empty, there is at least one packet arrival at time $t = 0$, and the buffer empties again for the first time at the end of slot $t = k$. Equivalently, we assume that $\mathcal{B}(0) = 0$, $\mathcal{B}(k) = 0$, and that there are at least $\lfloor Ct \rfloor$ departures from the buffer in the interval $[1, t]$ for any $t \in [1, k]$. It is enough to prove that $\mathcal{B}(t) \leq B + 1$ for all $t \in [1, k]$, since this would guarantee that no packets are dropped during intervals where the buffer is nonempty (and therefore no packets are dropped in general).

We let $a(t)$ be the total number of packet arrivals during slot t. Then for any $t \in [1, k]$ we have

$$\mathcal{B}(t) = \sum_{\tau=0}^{t-1} a(\tau) - \lfloor Ct \rfloor. \tag{11}$$

Since $\Lambda^T(t) \leq C$ for all t, we have that $\sum_{m=\tau}^{\tau+T-1} a(m) \leq \lfloor CT \rfloor$ for all τ. Therefore, there are at most $\lfloor t/T \rfloor \lfloor CT \rfloor$ packet arrivals during the interval $[0, \lfloor t/T \rfloor T)$, and at most $\lfloor CT \rfloor$ packet arrivals during the interval $[\lfloor t/T \rfloor T, t)$. This combined with Eq. (11) gives

$$\mathcal{B}(t) \leq \lfloor t/T \rfloor \lfloor CT \rfloor + \lfloor CT \rfloor - \lfloor Ct \rfloor \leq \frac{t}{T} CT + CT - Ct + 1 \leq CT + 1 \leq B + 1.$$

Q.E.D.

When the condition of Lemma 1 is violated, packets may be dropped due to buffer overflow. The following theorem gives a general upper bound on the packet loss probability on link $\mathcal{L}$ and is one of the main results of this discussion.

Theorem 1: The packet loss probability p satisfies

$$p \leq \frac{1}{\beta} \int_{v=C/NR}^{1} \left(\frac{\beta^v (1-\beta)^{1-v}}{v^v (1-v)^{1-v}} \right)^N \mathrm{d}v, \tag{12}$$

where R is the peak $\lfloor B/C \rfloor$-sliding-window rate of a session, r is the average rate of a session, and $\beta = r/R$.

Proof: We denote by $L(C)$ the average rate at which packets are dropped when the capacity of the link is C. To find an upper bound on $L(C)$, note that if from the incoming stream of packets we drop enough packets so that the condition $\Lambda^{\lfloor B/C \rfloor}(t) \leq C$ is satisfied for all t, then, as Lemma 1 suggests, no other packets will have to be dropped. Therefore, the average rate $L(C)$ at which packets are dropped satisfies

$$L(C) \leq E\left[(\Lambda^{\lfloor B/C \rfloor}(t) - C) 1_{\{\Lambda^{\lfloor B/C \rfloor}(t) > C\}} \right], \tag{13}$$

where $1_{\{x>a\}}$ is equal to one if $x > a$, and zero otherwise. Equation (13) gives

$$\begin{aligned} L(C) &= \int_{\omega=C}^{NR} (\omega - C) \Pr(\Lambda^{\lfloor B/C \rfloor}(t) = \omega)\, \mathrm{d}\omega = -\int_{\omega=C}^{NR} (\omega - C) \mathrm{d} \Pr(\Lambda^{\lfloor B/C \rfloor}(t) \geq \omega) \\ &= -\left[(\omega - C) \Pr(\Lambda^{\lfloor B/C \rfloor}(t) \geq \omega) \right]_{\omega=C}^{\omega=NR} + \int_{\omega=C}^{NR} \Pr(\Lambda^{\lfloor B/C \rfloor}(t) \geq \omega)\, \mathrm{d}\omega \\ &= \int_{\omega=C}^{NR} \Pr(\Lambda^{\lfloor B/C \rfloor}(t) \geq \omega)\, \mathrm{d}\omega, \end{aligned}$$

where we have used the fact that $\Pr(\Lambda^{\lfloor B/C \rfloor}(t) > NR) = 0$. The preceding equation together with Eq. (9) gives

$$L(C) \leq NR(1-\beta)^N \int_{v=C/NR}^{1} \left(\frac{1}{1-v}\right)^N \left(\frac{1}{v}-1\right)^{Nv} \left(\frac{\beta}{1-\beta}\right)^{Nv} \, dv,$$

where we have used the substitution $v = \frac{\omega}{NR}$, the definitions of R and β, and the fact that $\Pr(\Lambda^{\lfloor B/C \rfloor}(t) > NR) = 0$. The packet loss probability p is equal to the average rate at which packets are dropped over the total average packet arrival rate. Thus,

$$p = \frac{L(C)}{NR\beta},$$

which after some algebraic manipulation gives Eq. (12). **Q.E.D.**

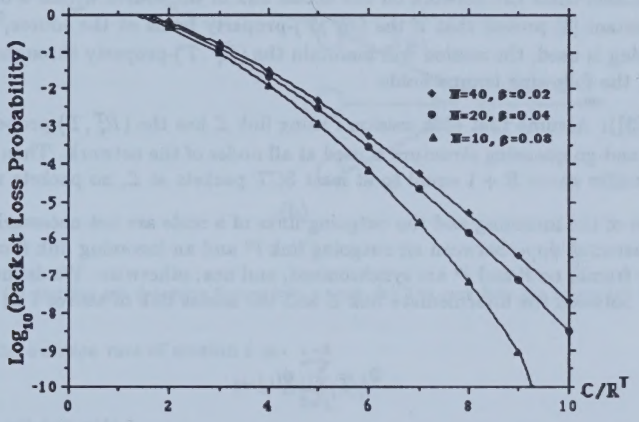


Figure 3: Upper bound on the packet loss probability as a function of C for different values of N and β^T.

The upper bound of Theorem 1 was proven without making any assumption about the particular probabilistic characteristics of the source arrival processes. The bound holds for any choice of the probabilistic characteristics of the arrival processes (which may be different for different sessions), and it depends only on the average rate and the peak ($\lfloor B/C \rfloor$)-sliding-window rate of the sessions, the number N of sessions, and the capacity C allocated to the sessions. It also depends (indirectly, through the definition of the peak rate) on the buffer size B.

5. AN UPPER BOUND THAT TAKES INTO ACCOUNT THE SESSION INTERACTIONS

The derivation of the upper bound on the packet loss probability in Section 4 was based on the assumption that the T-sliding-window rates $\lambda_i^T(t)$, $i = 1, 2, \ldots, N$, of the sessions sharing link $\mathcal{L}$ are independent. Although the assumption of independent sources is usually valid in practice at the entry points of the network, the independence is lost when the session paths intersect prior to their arrival at an intermediate link. The interactions between the sessions have the following consequences:

(a) The T-sliding-window rates $\lambda_i^T(t)$, $i = 1, 2, \ldots, N$, become correlated.

(b) The peak T-sliding-window rates R_i^T at the intermediate link may be smaller or larger than the corresponding peak rates at the sources (see [3]).

(c) The average rate r_i of session i arrivals at the intermediate link $\mathcal{L}$ may be smaller than the corresponding average rate at the source due to the dropping of packets at previous links.

We will neglect the effect of the packets that are dropped at links prior to link $\mathcal{L}$. By doing so, we can only overestimate the packet loss probability on link $\mathcal{L}$, and, therefore, any upper bounds that we obtain for the packet loss probability on $\mathcal{L}$ will still be valid. In what follows, we will modify our model in order to take into account the effects (b) and (c) above that result from the session interactions.

To do so we will assume that the packets of each session are transmitted by using the stop-and-go queueing structure proposed in [3]. In the stop-and-go queueing structure, the time axis of the incoming, outgoing, and access links of a node, is divided into time frames of fixed duration equal to T slots (see Fig. 4a). A packet arriving during frame F' of the incoming link l' of a node (or over an access link if the node is a source) is transmitted in the first frame F of the outgoing link l that starts after the end of F'. We will say that a source i has the $(\hat{R}_i^T, T)$-*property* if at most $\hat{R}_i^T T$ packets enter the network on the access link of the source within a frame of duration T slots. Golestani [3] proved that if the $(\hat{R}_i^T, T)$-property holds at the source, and if the stop-and-go queueing is used, the session will maintain the $(\hat{R}_i^T, T)$-property throughout the network. Furthermore, the following lemma holds.

Lemma 2 ([3]): Assume that each session i using link $\mathcal{L}$ has the $(\hat{R}_i^T, T)$-property at its source and the stop-and-go queueing structure is used at all nodes of the network. Then if $\sum_{i=1}^{N} \hat{R}_i^T \leq C$ and there is buffer space $B+1$ equal to at least $3CT$ packets at $\mathcal{L}$, no packets are dropped.

The frames of the incoming and the outgoing links of a node are not necessarily synchronized. The *phase mismatch* $\phi_{l',l''}$ between an outgoing link l'' and an incoming link l' of a node is equal to zero, if the frames on l' and l'' are synchronized, and one, otherwise. We define the total phase mismatch Φ_i between the intermediate link $\mathcal{L}$ and the access link of source i as

$$\Phi_i = \sum_{j=0}^{h-1} \phi_{l_j, l_{j+1}},$$

where l_0 is the access link of source i, $l_h = \mathcal{L}$, and $l_1, l_2, \ldots, l_{h-1}$ are the intermediate links on the path from source i to link $\mathcal{L}$ (see Fig. 4b). We denote by $G_i^T(k)$ the total number of packets that enter the network at source i during the k^{th} frame of the access link l_0, and by $E_i^T(k)$ the number of packets that are eligible for transmission during the k^{th} frame on link $\mathcal{L}$. The only packets of source i that may be eligible for transmission during frame k of link $\mathcal{L}$ are those that entered the network during frame $k - \Phi_i$ of the access link l_0. Thus, we have

$$E_i^T(k) \leq G_i^T(k - \Phi_i). \tag{14}$$

Equation (14) holds as a strict inequality when some of the packets generated at source i are dropped before reaching link $\mathcal{L}$. However, the worst case in terms of the number of packets that are dropped at link $\mathcal{L}$ corresponds to the case where Eq. (14) holds as equality. Since we are interested in upper bounds on the packet loss probability on link $\mathcal{L}$, we will assume, without loss of generality, that $E_i^T(k) = G_i^T(k - \Phi_i)$. The total number of packets that are eligible for transmission over $\mathcal{L}$ during frame k is $\sum_{i=1}^{N} G_i^T(k - \Phi_i)$. Since the arrival processes at the sources are independent, so are the random processes $G_i^T(k)$, and their delayed versions $G_i^T(k - \Phi_i)$ (the Φ_i's are deterministic scalars).

We define the *T-fixed-window rate* of session i during frame k on link $\mathcal{L}$ as

$$\hat{\lambda}_i^T(k) = \frac{G_i^T(k - \Phi_i)}{T}.$$

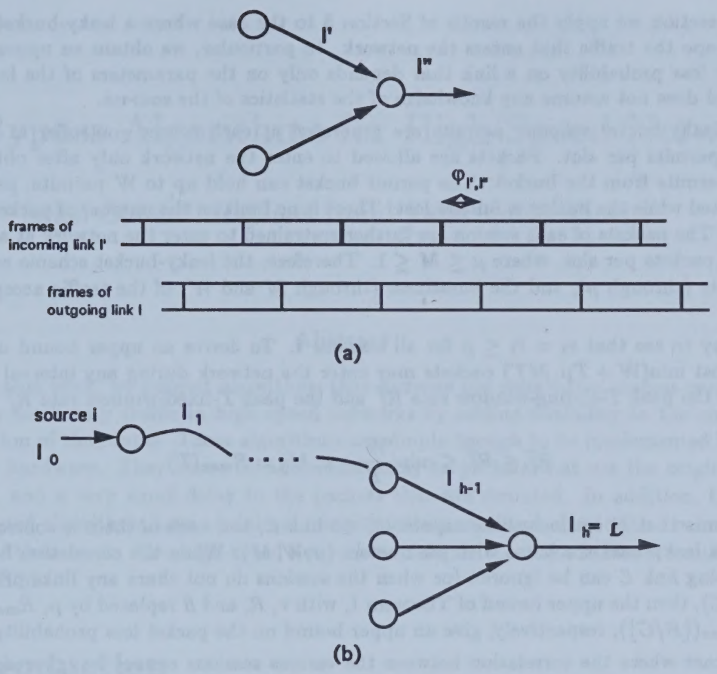


Figure 4: (a) Arriving and departing frames from a node. (b) The path from the access link l_0 to link $\mathcal{L}$.

We also define the *average rate* of session i as

$$\hat{r}_i = E[\hat{\lambda}_i^T(k)],$$

the *peak T-fixed-window rate*

$$\hat{R}_i^T = \max_k \hat{\lambda}_i(k),$$

and the *T-fixed-window smoothness coefficient*

$$\hat{\beta}_i^T = \frac{\hat{r}_i}{\hat{R}_i^T}.$$

Note that $\hat{r}_i = r_i$, $\hat{R}_i^T \leq R_i^T$, and $\hat{\beta}_i^T \geq \beta_i^T$, where the quantities r_i, R_i^T, and β_i^T were defined in Section 2.

Using Lemma 2 and an analysis similar to the one presented in Section 3, we can prove the following theorem.

Theorem 2: Consider a link $\mathcal{L}$ with buffer size $B+1$ on which N sessions are statistically multiplexed. Assume that each session has peak $\lfloor \frac{B}{3C} \rfloor$-fixed-window rate at most equal to $\hat{R}$, average rate at most equal to $\hat{r}$, and let $\hat{\beta} = \hat{r}/\hat{R}$. Then the packet loss probability p on link $\mathcal{L}$ satisfies

$$p \leq \frac{1}{\hat{\beta}} \int_{v=C/N\hat{R}}^{1} \left(\frac{\hat{\beta}^v (1-\hat{\beta})^{1-v}}{v^v (1-v)^{1-v}} \right)^N \mathrm{dv}. \tag{15}$$

6. BOUNDS ON THE PACKET LOSS PROBABILITY WHEN A LEAKY-BUCKET SCHEME IS USED

In this section we apply the results of Section 5 to the case where a leaky-bucket scheme is used to shape the traffic that enters the network. In particular, we obtain an upper bound on the packet loss probability on a link that depends only on the parameters of the leaky-bucket scheme and does not assume any knowledge of the statistics of the sources.

In the leaky-bucket scheme, permits are generated at each source controller at a constant rate of μ permits per slot. Packets are allowed to enter the network only after obtaining the required permits from the bucket. The permit bucket can hold up to W permits; permits that are generated while the bucket is full are lost. There is no limit on the number of packets that can be stored. The packets of each session are further restrained to enter the network at a maximum rate of M packets per slot, where $\mu \leq M \leq 1$. Therefore, the leaky-bucket scheme restricts the average rate (through μ), and the burstiness (through M and W) of the traffic accepted in the network.

It is easy to see that $r_i = \hat{r}_i \leq \mu$ for all sessions i. To derive an upper bound on R_i^T note that at most $\min(W + T\mu, MT)$ packets may enter the network during any interval of T slots. Therefore, the peak T-sliding-window rate R_i^T and the peak T-fixed-window rate $\hat{R}_i^T$ satisfy

$$\hat{R}_i^T \leq R_i^T \leq \min(\frac{W}{T} + \mu, M) = R_{\max}(T).$$

We assume that N sessions share capacity C on link $\mathcal{L}$, and each of them is controlled at the source by a leaky-bucket scheme with parameters (μ, W, M). When the correlation between the sessions using link $\mathcal{L}$ can be ignored (or when the sessions do not share any links prior to their arrival at $\mathcal{L}$), then the upper bound of Theorem 1, with r, R, and β replaced by μ, $R_{\max}(\lfloor B/C \rfloor)$, and $\mu/R_{\max}(\lfloor B/C \rfloor)$, respectively, give an upper bound on the packet loss probability on $\mathcal{L}$.

In the case where the correlation between the various sessions cannot be ignored, note that the traffic entering the network at a source has the $(R_{max}(\lfloor B/(3C) \rfloor), \lfloor B/(3C) \rfloor)$-property. If the leaky-bucket scheme is combined with the stop-and-go queueing structure, this property is maintained at the intermediate links, and Theorem 2 with $\hat{r}$, $\hat{R}$, and $\hat{\beta}$ replaced by μ, $R_{\max}(\lfloor B/(3C) \rfloor)$, and $\mu/R_{\max}(\lfloor B/(3C) \rfloor)$, respectively, gives an upper bound on the packet loss probability on $\mathcal{L}$.

REFERENCE

[1] Awerbuch, B., Cidon, I., Gopal, I., Kaplan, M., Kutten, S., "Distributed control for PARIS," technical report.

[2] Brassil, J. T., *Deflection Routing in Certain Regular Networks*, Ph.D. Thesis, University of California San Diego, 1991.

[3] Golestani, S. J., "Congestion-Free Communication in High-Speed Packet Networks," *IEEE Trans. on Communications*, Vol. 39, No. 12, December 1991.

[4] Greenberg, A. G., and Goodman, J., "Sharp Approximate Models of Adaptive Routing in Mesh Networks," in J. W. Cohen, O. J. Boxma and H. C. Tijms (Eds.), *Teletraffic Analysis and Computer Performance Evaluation*, pp. 255-270, Elsevier, Amsterdam, 1988.

[5] Greenberg, A. G., and Hajek, B., "Deflection Routing in Hypercube Networks," IEEE Trans. Communications, Vol. COM-35, No. 6, pp. 1070-1081, June 1992.

[6] Varvarigos, E. A., and Bertsekas, D. P., "A Conflict Sense Routing Protocol and its Performance for Hypercubes," accepted to the *IEEE Trans. on Computers*.

Fast Bypass Algorithms for High-Speed Networks

Israel Cidon*†, Raphael Rom†*‡ and Yuval Shavitt†

Abstract

In this work, we suggest algorithms that increase the reservation success probability for bursty traffic in high speed networks by adding flexibility to the construction of the routes. These algorithms are simple enough to be implemented by cheap hardware. They cause no additional delay to packets that use the original route, and a very small delay to the packets that are rerouted. In addition, the presented algorithms have a minimal communication overhead, due to the local nature of their work. Two high-speed network models are considered: source routing and ATM.

1 Introduction

High speed networks are intended to support applications with widely varying traffic characteristics: from short database queries to long video streams. In order to use the network resources efficiently, bandwidth reservations are made to ensure high probability of data arrival to their destinations. For applications such as constant bit rate video or voice conversations this is the right approach. However, for bursty traffic, i.e., traffic whose intensity varies in time, reservation itself introduces non-negligible overhead. Moreover, the widely varying nature of bursty traffic indicates that a simplistic burst reservation mechanism would not suffice. The scheme must consider the burst size and the timing constraints in its operation, as we briefly explain below.

Short bursts are those whose transmission time is not more than a few round trip delays. For such bursts waiting for a reservation, that itself takes a few round trip delays, is clearly not acceptable. The best method for this type of bursts is to make an initial zero-bandwidth reservation and subsequently to send the data without reservation and use time-outs (possibly at a higher layer) to detect failures. Turner [10] suggested an on-the-fly reservation scheme. In his scheme a burst that arrives to an ATM switch and finds sufficient bandwidth for its cells, reserves the required bandwidth (to prevent new bursts from disturbing this one) and proceeds to the next switch towards its destination. This scheme does not guarantee that a burst that succeeds in reserving enough bandwidth in one switch will also succeed in the next one along the route. Hence the choice of the route is crucial in the success of the on-the-fly reservation.

*Sun Microsystems, Mountain View, CA 94043-1100

†Department of Electrical Engineering, Technion, Haifa 32000, Israel

‡This work was supported in part by the Technion V.P.R. fund and the Fund for the Promotion of Research at the Technion.

The same solution does not fit longer bursts. Here, the overhead of reservation is not as bothersome so traditional reservation algorithms can be used. Note, however, that this approach is valid only if there is enough storage at the source to hold the burst data until positive acknowledgment is received for the reservation signaling [2, 5]. Thus, such an approach would be useful for bursty data applications such as FTP in which the data can be easily kept in the source. This approach would not be useful for bursty real-time application, e.g., variable bit rate video, which (for storage reasons) cannot tolerate long waiting times for a reservation process to complete.

In this work we suggest algorithms that increase the probability to successfully transfer bursty traffic by adding flexibility to the burst routing. We assume that bursty applications reserve no bandwidth during their set-up process. Instead, bandwidth is requested for each burst separately (with either on-the-fly or traditional fast reservation algorithms) and is freed immediately after the burst transmission. The suggested bypass algorithms are simple enough to be implemented by cheap hardware. Before proceeding we describe two routing approaches for high speed networks with which our algorithms can be used: source routing and ATM.

Source routing, or *Automatic Network Routing (ANR)* [3], is a routing method where each packet carries in it the entire route it should traverse. In our discussion, we will assume that the route is placed in the header as a list of port-IDs (or link-IDs), and each node along the packet route strips the ID it uses from the head of the list (in practice, there are other methods for handling the source route that only differ in technicalities and can be integrated with our algorithm [3]). In networks that employ ANR, the route for the session is computed at the source node using data that is distributed by a *topology update* algorithm. It is therefor plausible that routes thus computed are not optimal (and may not even be feasible). Changing the route on-the-fly amounts to modifying the source route in the packet's header.

In ATM networks, cells travel along Virtual Circuits (VCs) that are constructed by a concatenation of Virtual Paths (VPs). The VC and VP identifiers are written in the cell header and possibly swapped in every switch. Tables in the switches are used to determine the route based on local identifiers [1]. For our purpose it is important to note that the routing information is distributed in the switches along the path the cells traverse. Modifying a cell's route on the fly requires changing the routing information in several switches – an operation that is neither simple nor fast [8]. In particular, buffering requirements for the cells while a new route is created makes on-the-fly rerouting look impractical.

The algorithms we suggest in this work increase the probability of a successful short burst transmission or the probability of a successful reservation for longer bursts by using local route-deflections. Because the route is determined based on somewhat inaccurate data, and because a proper reservation process is not undertaken, it is possible that the determined route may actually not be able to accommodate the bandwidth of the burst. To overcome this possible lack of bandwidth local route deflections are constructed. To use these deflections our algorithms use load information from the immediate neighboring nodes. This does not require dissemination of large volumes of load-data across the network, keeps the information fairly up to date, and increases the probability of reservation success.

Most high speed networks are constructed as an interconnection of specially constructed packet switches. Unlike traditional switches these switches must support ex-

tremely fast streams of small cells meaning that switching speed is very high and implying that the use of a software operated general-purpose processor is out of the question. A typical switch is constructed as an interconnection of *link processors* (LPs) each supporting a single link [3, 4]. The routing of packets that arrive at the input links is done directly by these LPs. Only packets that require more complex processing (e.g., control packets) are forwarded to a more sophisticated control unit. Naturally, the suggested algorithms are designed to be performed by the LPs.

The rest of the paper is organized as follows. In section 2 we describe a fast bypass algorithm for networks that employ ANR. Next we describe in section 3 a fast bypass algorithm for ATM networks. In section 4 we analyze the performance of the two algorithms in terms of reservation success probability, and in section 5 we conclude.

2 A Fast Bypass Algorithm for Networks with ANR

The LPs in an ANR switch share the routing tables that are used for the presented algorithm. Thus, it is convenient to treat the switch LPs (the hardware) and the controlling algorithms as a single abstract entity, the *node*. The algorithm we present in this section works by exchanging load information in a close locality of every node. To that end, we make use of the following definitions. The *2-neighborhood* of a node is the set of nodes that are at most two hops away. A *local segment* is a single link or a concatenation of two links leading from a node to another node in its 2-neighborhood. A local segment is typically a part of a longer path between source and destination nodes. Note that every local segment uniquely identifies a node, but several local segments may identify the same node. The *local segment group* is the collection of all the local segments that identify the same node. We use (l_1, l_2) to signify a two-hop local segment comprised of links l_1 and l_2 (in that order), and (l_1) for a one-hop local segment. The bypass algorithm will, in congested situations, replace one local segment with another.

The Bypass Algorithm, *BA*, is based on frequent load measurements (typically the load indicator is the sum of the total reserved bandwidth and the average number of buffers occupied by nonreserved traffic) of the outgoing links at every node, and on sharing this information with the immediate neighbors. This way every node has updated knowledge of the load on all the local segments emanating from it. This information is maintained in a local *RoutingTable* that has an entry for each local segment. Each entry in the *RoutingTable* contains a field that indicates the availability of the local segment, and the preferred alternate local segment should the original one be blocked (see below). *RoutingTable* size is quadratic in the output degree of the nodes, and in most practical networks is not expected to have more than a few tens of entries.

Figure 1A shows an example of a five node network. The routing table of node S has 10 entries for the following local segments: (4) and (3, 5) to node B; (4, 5), (3), and (1, 2) to node D; (1), (3, 2), and (6, 7) to node I; and (6) and (1, 7) to node C. Suppose a packet arrives to node S with the route (1, 2) written in its header and finds link 1 blocked. Three types of bypasses are possible for this packet (see figure 1B): a) the direct link to D (a shortcut) that avoids both links 1 and 2, b) a two-link bypass via node B (identical length) that again avoids links 1 and 2, or c) a two-link bypass of link 1 to node I (a long bypass) that is followed by link 2. These types are the only ones considered in this algorithm.

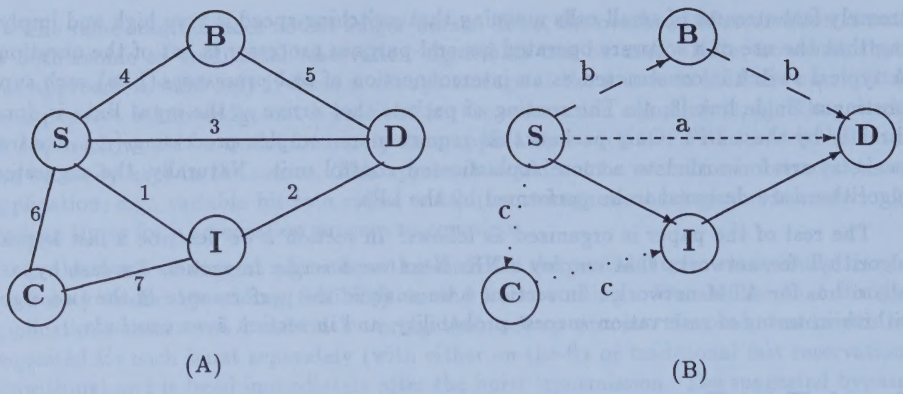


Figure 1: An example network with the three bypass types

When a packet arrives at a switch, its entry in *RoutingTable* is examined to check if the local segment it should traverse is not blocked. If it is blocked *RoutingTable* is first searched for a bypass that does not increase the length (types a and b) and then for one that bypasses only this link (bypass type c). If the search succeeds the new local segment replaces the original one in the packet header and the packet is forwarded along the deflected route; otherwise, the packet is discarded.

To maintain *RoutingTable*, we keep for every entry the load of the preferred route. An entry in *RoutingTable* is updated when the load on the preferred route is changed, or when a bypass route with more residual bandwidth (lower load) than the preferred route is found. Link failures are treated as a maximal decrease in the available bandwidth as will be explained in section 2.1.

To expedite the processing of regular packets *RoutingTable* is sorted according to the local segments. This, however, poses an update problem since there may be several local segments that identify the same node and it is natural to maintain their bypass information simultaneously. To allow simultaneous updates of all the entries in *RoutingTable* that refer to the same node, a second table, *HostTable*, is used. *HostTable* is sorted by node-IDs[1] with a single entry for every node in the 2-neighborhood. Each entry contains the node-ID and a list of all the local segments that identify this node, i.e., its local segment group. *HostTable* can be initialized either when the network is started or can be built by a topology update algorithm [9].

2.1 A Detailed Description

Two tables are maintained and used by the algorithm: *RoutingTable* has an entry for each local segment that comprises four fields: • The id of the node at the end of the local segment. • The load of the local segment. • The preferred alternative local segment. • The load of the preferred alternative local segment. The maintenance of this table is described below. *HostTable* has an entry for each node in the 2-neighborhood that lists all the local segments in the *local segment group* of the node.

[1]Node IDs can be global or locally assigned by higher level algorithms.

Every node periodically measures the loads on the links that emanates from it. The way these measurements are made is out of the scope of this paper. For our purpose, it is enough to assume that the resulting load indicator is based on both reserved and non-reserved traffic. The load indicators are sent to the immediate neighbors and are locally used to update *RoutingTable* as follows. For every emanating link, the load entry of the one-hop local segment is updated. If the direct link is the preferred route then the preferred load field in the entries of its local segment group are updated. If its load is lighter than the one of the preferred route of its local segment group then the local segment is written in the preferred route field and its load is written in the preferred route load field in the entries of all the members of the local segment group. The members of the group are easily located with *HostTable*.

The measurements are sent to the neighbors as a list of number pairs, a link id and its load. For every link in the list, the load of the corresponding local segment is updated. If this local segment is the preferred local segment the load field of the preferred route in the entries of all the members in the local segment group is updated. If the local segment load is lighter than the one of the preferred local segment then the first becomes the preferred local segment and the entries of the local segment group are changed to reflect this, i.e., the local segment is written in the preferred local segment field and its load is written in the preferred local segment load field.

A failure in a link that is not directly connected to a node is treated as if the available bandwidth of this link dropped to zero, and can be reported by sending a measurement list. A failure in a link adjacent to a node requires a pass through the entire *RoutingTable* (typically, few tens of entries) to search for all the entries that have this link as part of their preferred local segment, and then to update their load to the maximum, so that every new measurement of a different local segment will update it. This process is not efficient, but is used only in the rare event of link failure and only in the two nodes at the ends of the failed link.

2.2 Avoiding Loops

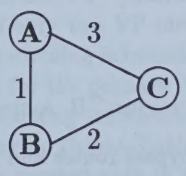


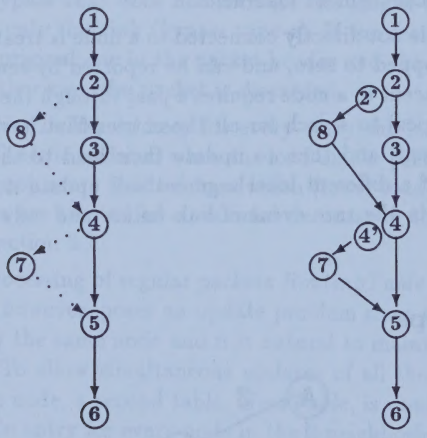
Figure 2: An example of a bypass loop

Deflection routing in general and *BA* in particular may cause a packet to cycle in loops. We next demonstrate how *BA* can cause a message to travel in a two link loop. Consider the network of Figure 2 and suppose a packet reaches node A and is trying to reach node C through link 3. At the time the packet arrives at node A, the buffers of link 3 are all full, there are free buffers in front of link 1, and *RoutingTable* indicates that the preferred bypass for the local segment (3) is the local segment $(1,2)$. The packet is, thus, sent via the bypass $(1,2)$. Suppose the packet is somewhat delayed in the queue and when it reaches node B link 2 has no free buffers but according to *RoutingTable* the

preferred bypass to segment (2) is (1,3) since links 1 and 3 are not totally full now. As a result the packet is deflected back to node A.

On the one hand, going in circles or busy waiting, can be considered as a good solution since instead of discarding the packet we use the network as storage. On the other hand, if the buffers of the switches are all almost full we might create a livelock where messages travel around and never reach their destinations, or do so after consuming too much network resources. To disable routing loops, a bit in the packet header can be set the first time the packet is deflected and if a second deflection is needed the packet is discarded. If more routing flexibility is needed few bits can be allocated in the packet header to bound the number of deflections above one. Two or more allowed deflection may theoretically cause a packet to go in cycles, but in practice, the probability for this is small if the number of bypasses is kept small. The analysis in section 4 shows that allowing only one deflection significantly decrease the rejection probability, while the residual contribution of additional deflections to the success probability of the packet decrease for every additional allowed bypass.

3 A Fast Bypass Algorithm for ATM Networks



A. A VP with bypass routes. B. A tree representation of A.

Figure 3: A VP with bypass routes and its tree representation

In this section we adapt the bypass algorithm presented in the previous section to ATM networks. In networks that employ ANR, once we identify a blocked link and know about a local segment that bypasses it, we can deflect the message by changing the routing information in its header. In ATM networks the routing information is not carried in the cells but is scattered in the switches along the path it traverses. Changing the routing information in several switches to create a deflection route is therefore neither simple nor fast [8]. In particular, buffering requirements for storing the cells while the new route is created make on-the-fly deflection look impractical.

We suggest, instead, to preprepare for congestion when a VP is constructed. Upon a VP construction, loaded areas are identified and bypass routes are created to be used

when the primary route is blocked. Each VP_i is assigned two VPIs: VPI_i for the primary route and VPI'_i for the bypass route. In figure 3A the primary VP is drawn with solid lines and the bypass routes with dotted lines. As already stated, bursty VCs are brought up with no bandwidth reservation, and use a fast reservation algorithm whenever there is a burst to transmit [2]. In ATM networks, fast reservation algorithm use monocell messages that traverse the VC route, reserve bandwidth in one direction, and acknowledge/reject a reservation request in the reverse direction. Since ATM VCs are unidirectional, the reverse direction does not necessarily use the same physical links.

We now show how reservation cells are deflected to create bypass VCs, and how the switches identify and route cells that belong to bypass VCs. We term the switches where a bypass starts (switches 2 and 4 in figure 3A) *junction switches.* A reservation cell starts its way on the VP with the primary VPI. Non-junction switches do not participate in the deflection process. When a reservation cell arrives at a junction switch that is unable to fulfill the burst request for bandwidth in the primary route it tries to reserve bandwidth in the first link of the bypass. Upon a success, the reservation cell is forwarded to the bypass link and its VPI is set to the secondary VPI. In addition, the junction switch registers the VC in a separate table, *BypassTable*. The values registered in *BypassTable* are the secondary VPI (cells of this burst will use it as their VPI), the VCI, and the rest of ATM switching data, i.e., the local VPI in the next hop and the output port-ID. A bandwidth release cell that follows the end of the burst cause the deletion of the corresponding entry from *BypassTable*.

Usually, the arrival of a reservation cell to the switch at the end of the VP triggers the transmission of an *Ack* to the VC-switch at the beginning of the VP. Similarly, the successful arrival of a reservation cell with the secondary VPI to the end of the VP triggers the transmission of a similar message, *Ack'*, signaling the local source that the cells of this burst should be switched through the secondary VPI (or, if the source is the origin of the cells, it should initiate the cells with the secondary VPI). When a data cell arrives with the secondary VPI at a junction switch, *BypassTable* is searched and if a match is found the cell is routed according to the data in the table. Otherwise, if no match is found in *BypassTable* the cell is routed according to the ATM switching table.

The suggested bypass algorithm offers 2^m potential routes (where m is the number of junction switches) for the price of only two VP identifiers and less than four switching table entries in every switch. The switching information to the next switch in the primary route is saved in two entries: one for the primary VPI and one for the secondary VPI. The two entries have identical switching information. If a VC reserves bandwidth for a burst in a bypass route, an entry in *BypassTable* keeps the switching information of the deflected route. This entry is created only in the junction switches where the VC is deflected, and is deleted when the bandwidth is freed. In the switches of the bypass routes one entry for the secondary VPI is kept in the regular switching tables. There is no need for an entry for the primary VPI.

In practice, a network manager might wish to bound the number of bypasses to keep the stretch factor, i.e., the ratio between the original VP length (in hops) and the length of the VC with the bypasses, low. A small counter in the reservation cell can be used to implement any practical bound, and in particular one bit can be used to allow only one bypass.

Examining the VP with the bypasses of figure 3A one can easily identify a tree rooted at the destination as depicted in figure 3B. This suggests an alternative way to look at the

bypass scheme: instead of building a shoelace VP, we build a VP with a tree structure, such that every leaf except the VP entry point must also be an internal node of another branch of the tree. An interesting extension to this algorithm will enable to bypass a bypassed route, e.g., in figure 3 if the link between switches 7 and 5 is loaded one may wish to use a direct link between switches 7 and 6. Other uses of tree shaped VPs can be found in [8].

4 Analysis

In this section we compute the improvement in the reservation success probability when our algorithm is used in triangulated graphs. Other regular structures, hypercubes, grids, and chordal ring, are considered in [7]. Since the improvement is not identical for the ANR and ATM networks we point out these differences. We assume that the probability to succeed in reserving bandwidth of a link is p for all the links, and this probability is independent for every link. The independence is justified by the fact that a burst is mainly competing against other applications that have constant bit rate, and not against other bursts. We assume that the original routes are all shortest path routes.

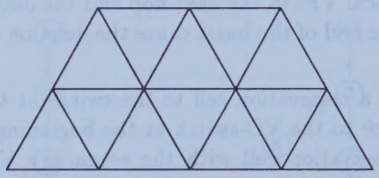


Figure 4: A lattice of triangles

A planar graph where every region is bounded by a circuit of three edges is said to be *triangulated* [3]. Consider, first, a lattice of triangles (figure 4). A shortest path route in a triangular lattice has no 60° turns. The probability that a two-hop segment is congested is $1-p^2$.

We look, first, at networks that employ ANR. If there are no turns in a path, a congested link can be bypassed by one of two possible two-link type c bypasses with probability $1-(1-p^2)^2$. If there is a 120° turn in the path the bypass probability is higher since an additional type b bypass can be found. When only one bypass is allowed, the success probability of reservation along an h hop route is at least $p^h\left(1+h(1-p)p(2-p^2)\right)$.

We assume that in an ATM network only one bypass route is prepared per link. The success probability under this assumption when only one bypass is allowed is at least $p^h\left(1+h(1-p)p\right)$.

If K bypasses are allowed the success probability is $p^h\sum_{k=0}^{K}\binom{h}{k}(1-p)^k p^k$. For $K=h$ we get $p^h\left(1+p-p^2\right)^h$.

Figure 5 compares the probability for a burst to succeed in reserving bandwidth along its VC in an ATM network as a function of the probability to find enough bandwidth on a link. The solid line depicts the success probability without bypasses, the dashed line depicts the case when only one bypass is allowed, and the dotted line depicts the case when there is no restriction on the number of bypasses. It is clear from the graphs that the effectiveness of the bypass algorithms grows with the length of the VP. Note

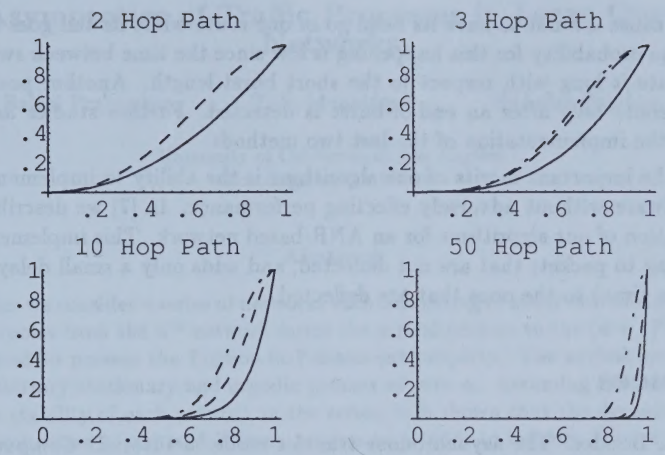


Figure 5: The effect of the Bypass Algorithms on the success probability of bursts as a function of the link load

also that even when only one bypass is allowed the improvement is significant, in [7] we demonstrate that the contribution of more bypasses decreases with each additional allowed bypass. This may lead us to implement the algorithm with the one bypass option which is the easiest and the most efficient implementation.

For general triangulated networks, let h be the number of links in a path of length H that are part of a triangle. Note that since we consider only shortest path VPs, no two links in a route belong to the same triangle. Based on our previous results, $p^H(1+h(1-p)p)$ is a lower bounds for the success probability when only one bypass is allowed (In the worst case there is only one type C bypass for each of the h links in both network models). If K bypasses are allowed the success probability is $p^H \sum_{k=0}^{K} \binom{h}{k}(1-p)^k p^k$. For $K=h$ we get $p^H(1+p-p^2)^h$.

5 Concluding Remarks

The algorithms presented here can be used in both models, ATM and ANR, for short bursts that may use on-the-fly reservation and for longer bursts that use fast reservation algorithms. For bursts that use fast reservation, our algorithms adhere to the reservation principle of the network whether it is ATM based or ANR type. For non-reserved traffic, one should be cautious not to allow this traffic to disturb the reserved traffic. This can be achieved by setting a low priority bit of the non-reserved traffic cells (in ATM the CLP bit; in the ANR model, such a bit should be allocated in the packet header).

Deflection of short bursts that do not reserve bandwidth before transmission in ATM networks can be done in two distinct ways: with or without reservation on-the-fly. If on-the-fly reservation is used each burst is preceded by a reservation cell and succeeded by a release cell. The cells are sent with VPI', the secondary VPI, and are treated as described in section 3 for bursts that use reservation, i.e., only if no bandwidth is available in either route the burst is discarded. If no reservation is used, non-reserved cells should be identified and switched to the primary or the secondary route regardless of their VC.

This might cause a burst to have its head go in one route while its tail goes in the other. However, the probability for this happening is low since the time between switching from route to route is long with respect to the short burst length. Another possibility is to switch the route only after an end-of-burst is detected. Further studies are needed to investigate the implementation of the last two methods.

One of the important merits of our algorithms is the ability to implement them with simple hardware without adversely effecting performance. In [7] we describe a possible implementation of our algorithms for an ANR-based network. This implementation does not add delay to packets that are not deflected, and adds only a small delay (a few byte transmission time) to the ones that are deflected.

References

[1] J.-Y. L. Boudec. The asynchronous transfer mode: a tutorial. *Computer Networks and ISDN Systems*, 24:279 – 309, 1992.

[2] P. E. Boyer and D. P. Trachier. A reservation principle with applications to the ATM traffic control. *Computer Networks and ISDN Systems*, 24:321 – 334, 1992.

[3] R. G. Busacker and T. L. Saaty. *Finite Graphs and Networks: an introduction with applications.* McGraw-Hill, 1965.

[4] I. Cidon and I. Gopal. PARIS: an approach to integrated high-speed private networks. *Intl. J. of Digital and Analog Cabled Systems*, 1(2):77 – 86, Apr.-June 1988.

[5] I. Cidon, I. Gopal, P. M. Gopal, R. Guérin, J. Jannielo, and M. Kaplan. The plaNET/Orbit high speed network. Technical Report RC-18270, IBM, T. J. Watson Research Center, Yorktown Heights, NY, March 1993.

[6] I. Cidon, I. Gopal, and A. Segall. Fast connection establishment in high speed networks. In *ACM SIGCOM'90*, pages 287 – 296, 1990.

[7] I. Cidon, R. Rom, and Y. Shavitt. Fast bypass algorithms for high-speed networks. Technical Report EE PUB No. 924, Technion - Israel Inst. of Tech., June 1994.

[8] R. Cohen, B. Patel, F. Schaffa, and M. Willebeek-LeMail. The sink tree paradigm: Connectionless traffic support on ATM LANs. In *INFOCOM'94*, pages 821 – 828, June 1994.

[9] R. Cohen and A. Segall. Connection management and rerouting in ATM networks. In *INFOCOM'94*, pages 184 – 191, June 1994.

[10] A. Segall. Distributed network protocols. *IEEE Transaction on Information Theory*, IT-29(1):23 – 35, January 1983.

[11] J. S. Turner. Managing bandwidth in ATM networks with burtsy traffic. *IEEE Network*, 6(5):50 – 58, September 1992.

The Asymptotics of Traffic Processes in Large Queueing Networks

Balaji Prabhakar T. S. Mountford Nicholas Bambos

University of California at Los Angeles
Los Angeles, CA 90024

Abstract

In this paper we consider a series of networks each consisiting of many individual queueing nodes. The departure process from the n^{th} network forms the arrival process to the $(n+1)^{th}$ network. Each network is assumed to possess the Poisson-in-Poisson-out property. The arrival process to the first network is an arbitrary stationary and ergodic process of rate α. Assuming that α is small enough to guarantee the stability of each network in the series, it is shown that the departure process from the n^{th} network converges to a mixed Poisson process as n tends to infinity. For networks possessing special topologies (for example fully-connected, "star", etc.), it is shown that the departure process converges to a Poisson process. The class of networks for which these results apply includes Jackson Networks.

Departures from a series of $\cdot/\mathrm{GI}/\infty$ queues are studied using couplings for random walks and extensions of the result of Vere-Jones [11] are obtained.

1. Introduction

In this paper we consider a collection of networks J_n, $n \in \mathbb{Z}^+$, each of which is an interconnection of many individual queueing nodes. The networks are connected in a series fashion with the departures from J_n forming the arrivals to J_{n+1} (see Figure 1). The arrival process to the first network, $\mathbf{A}^1$, is an arbitrary stationary and ergodic process of rate α. The departure process from the first network, $\mathbf{A}^2$, is then fed to the second network and we obtain $\mathbf{A}^3$ as the departure process from J_2. Proceeding thus, we obtain $\mathbf{A}^n$ as the departure process from network J_{n-1}.

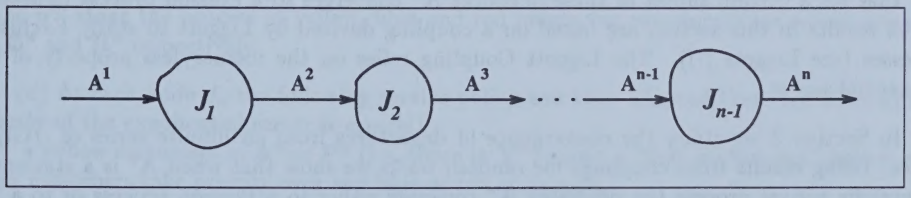


Figure 1: A Sequence of Networks

We assume that the service times of the various customers that arrive to a specific queueing node, are distributed i.i.d. (although the service time distribution of a specific customer at diffferent nodes may be different). The routing of customers inside each J_n is assumed to be

Markovian. Thus, there is no division of the customers into different classes based either on service-time or on routing requirements. We also suppose that the service rates at the individual stations inside each network are sufficiently high so that all the networks are stable. Thus, in equilibrium, each of the $\mathbf{A}^n$'s is a stationary and ergodic process of rate α. Futhermore, the J_n's possess the Poisson-in-Poisson-out property, i.e. if $\mathbf{A}^1$ is a Poisson process of rate α, then so is each $\mathbf{A}^n$, $n \in \mathbb{Z}^+$. These conditions are satisfied if, for example, each J_n is a Jackson Network with appropriate service rates.

We investigate conditions under which $\mathbf{A}^n$ converges in distribution to a Poisson process.

Our approach is based on the recent work of Mountford and Prabhakar [9] which established the following conjecture of Reiman and Simon [10]: the limit from passing an arbitrary stationary and ergodic process of rate $\alpha < 1$ through an infinite series of independent, rate 1, exponential-server $(\cdot/\mathrm{M}/1)$ queues is a Poisson process of rate α. One of the main ingredients of the proof of the above conjecture is the fact that the $\cdot/\mathrm{M}/1$ queueing node possesses the Poisson-in-Poisson-out property. Given this, it is natural to ask whether a similar result is true for other types of queueing systems which also possess the Poisson-in-Poisson-out property. Perhaps the most important examples of such queueing systems are Jackson Networks and Kelly Networks.

The problem of convergence of traffic processes in large networks has a long history and has been studied by many scholars. Investigating the flow of messages in multi-hop communication networks, Kleinrock [5] posed the question of characterizing the limiting distribution of the message flows as the number of hops becomes large. The result of Mountford and Prabhakar [9] stems directly from the works of Liggett and Shiga [8] and Anantharam [1]. Other contributions to the study of the limiting flows of departures in queueing networks from various perspectives may be found in the papers of Bambos and Walrand [2], Glynn and Whitt [4], and Vere-Jones [11].

In Section 2 we recall the salient features of the methods used in Mountford and Prabhakar [9]. Theorem 1 shows that the limit of departures from a series of networks J_n (as described above) is a mixed Poisson process. By imposing certain topological conditions on the J_n we then show that for a certain subset of these networks $\mathbf{A}^n$ converges to a Poisson process (Theorem 2). All results in this section are based on a coupling devised by Liggett to study Exclusion Processes (see Liggett [7]). The Liggett Coupling relies on the memoryless property of the exponential server.

In Section 3 we study the convergence of departures from an infinite series of $\cdot/\mathrm{GI}/\infty$ queues. Using results from couplings for random walks we show that when $\mathbf{A}^1$ is a stationary and ergodic arrival process the processes $\mathbf{A}^n$ converge either to a Poisson process or to a "ν-Poisson process" (defined later) depending upon the nature of the service distribution. The coupling used in this section depends on the fact that are an infinite number of servers and so we call this the I-S Coupling. It will be seen that the Liggett Coupling and the I-S coupling are complementary in that situations which cannot be analyzed using the one can be studied using the other.

2. Limiting Departures from a Series of Jackson Networks

All results in this section are based on the methods used in Mountford and Prabhakar [9]. We now recall the essentials of their argument. Suppose each of the J_n's of Figure 1 is a $\cdot/M/1$ queue with mean service rate 1 that operates independently of the other queues. Let $\mathbf{A}^1$ be a stationary and ergodic arrival process of rate $\alpha < 1$. The main idea of Mountford and Prabhakar [9] is to couple $\mathbf{A}^1$ with an independent, rate α, Poisson process $\mathbf{P}^1$ (the Liggett Coupling) and then show that $\mathbf{A}^n$ increasingly shares more points with $\mathbf{P}^n$. Since each $\mathbf{P}^n$ is Poisson, this would then imply that $\mathbf{A}^n$ converges to a Poisson process once it is shown that the intensity of common points increases upto α. The details are as follows.

(a) Let the *service process* of each of the J_n's be denoted by $\mathbf{N}_n$. The $\mathbf{N}_n$, $n \in \mathbb{Z}^+$ are an i.i.d. family of rate 1 Poisson processes, independent of the arrival process $\mathbf{A}^1$. Let $X_A^1(t)$ be the stationary and ergodic process representing the queue-size at the first queue at time t. The process $X_A^1(\cdot)$ is generated from the joint process $(\mathbf{A}^1, \mathbf{N}^1)$. The departure process from the first queue, $\mathbf{A}^2$, is then given by $\mathbf{A}^2(\omega) = \{t \in \mathbb{R} : t \in \mathbf{N}^1(\omega) \text{ and } X_A^1(t^-) > 0\}$ (see [9] for details). The process $\mathbf{A}^2$ is stationary and ergodic of rate α and forms the input to the second node, J_2. From $(\mathbf{A}^2, \mathbf{N}^2)$ we can then generate the queue-size process at the second node $X_A^2(\cdot)$ and the corresponding departure process $\mathbf{A}^3$. Proceeding thus, we may generate the processes $X_A^n(\cdot)$ and $\mathbf{A}^n$ from $(\mathbf{A}^{n-1}, \mathbf{N}^{n-1})$.

(b) Choose a Poisson process $\mathbf{P}^1$ of rate α, independent of the family $\{\mathbf{N}^n\}$ and of $\mathbf{A}^1$. Let $\mathbf{P}^n$ be the arrival process to the n^{th} node, J_n, when $\mathbf{P}^1$ is inputted to J_1. As in step (a) we may then obtain the processes $X_P^n(\cdot)$ and $\mathbf{P}^n$ from $(\mathbf{P}^{n-1}, \mathbf{N}^{n-1})$. Note that, by Burke's Theorem, $\mathbf{P}^n$ is Poisson for each $n \in \mathbb{Z}^+$. It is worth mentioning here that we fix $\mathbf{N}^n$ to be the service process associated with the node J_n *regardless* of the specific arrival process to J_n. This construction relies on the memoryless property of the service process $\mathbf{N}^n$ (see [9] for details).

(c) Couple the processes $\mathbf{A}^n$ and $\mathbf{P}^n$, and colour the points in $\mathbf{A}^n \cup \mathbf{P}^n$ according to the following rules

- points in $\mathbf{A}^n \cap \mathbf{P}^n$ are coloured yellow
- points in $\mathbf{A}^n$ but not in $\mathbf{P}^n$ are coloured blue
- points in $\mathbf{P}^n$ but not in $\mathbf{A}^n$ are coloured red.

At the n^{th} stage the process of yellow, blue and red points (or customers) are represented by $\mathbf{Y}^n$, $\mathbf{B}^n$ and $\mathbf{R}^n$ respectively.

(d) At each node J_n the following service policy was then adopted (here the memoryless property of the exponential server is crucial)

- yellow customers in $\mathbf{A}^n$ or $\mathbf{P}^n$ (i.e. those in $\mathbf{Y}^n$) adopt a "first in, first out" policy, and yellow customers take priority over any blue or red customers,
- an arriving blue customer of $\mathbf{A}^n$ who finds a red customer at the queue "coalesces" (or couples) immediately with the red customer who arrived first and has not yet coalesced. Both the "coalesced customers" will be coloured yellow in $\mathbf{A}^{n+1}$ and $\mathbf{P}^{n+1}$. Similarly with arriving red customers that find uncoalesced blue customers.

It should be pointed out here that as a consequence of the coupling and the service policy,

one immediately has the following.

- A yellow customer remains yellow forever, and so the density of yellow customers (which, by the ergodicity of $\mathbf{Y}^n$, is non-random) is non-decreasing.
- "Orderings" between blue and red customers are preserved. That is, if U belonging to $\mathbf{B}^n$ ($\mathbf{R}^n$) and V belonging to $\mathbf{R}^n$ ($\mathbf{B}^n$) are such that U arrives before V at node n then U will arrive before V at every successive node unless V becomes yellow at some stage. Hence, a blue (red) customer can never overtake a red (blue) customer unless she turns yellow.

(e) Based on an argument of Ekhaus and Gray [3] which exploits the "order-preserving property" mentioned above, it is shown in [9] that the density of yellow customers increases upto α and so $\mathbf{A}^n$ converges to a Poisson process in distribution.

It is now easy to see that a suitable modification of the service policy (without changing the input-output properties of the node) extends the argument in [9] to the following infinite systems and it can be seen that the departure processes converge weakly to a Poisson process. The basic idea in all of the cases below is to couple the arbitrary stationary and ergodic arrival process of rate α, $\mathbf{A}^1$, with a Poisson process as before and give priority to yellow customers. Whenever there is a routing of customers involved (routing is assumed to be Bernoulli), yellow customers are routed together while red and blue customers are routed on an i.i.d. basis.

1) **Infinite Series of $\cdot$/M/1 Feedback Queues**
2) **Infinite Series of $\cdot$/M/k ($\cdot$/M/∞) Queues**
2.b) **Infinite Series of $\cdot$/M/k ($\cdot$/M/∞) Feedback Queues**
3) **Infinite Sequence of $\cdot$/M/1/k Feedforward Queues:** Each node is precisely a $\cdot$/M/1 node (with service rate 1) except that the buffers are all finite and of size k. If an arrival finds the buffer full, it is immediately shunted to the output and thus gets instantaneous service. For any finite k, it is well-known that the equilibrium departure process is Poisson if the arrival process is Poisson of rate less than 1 (see Kelly [6], page 36).

2.1 Main Results for Jackson Networks

Referring to Figure 1 again, each J_n is now assumed to consist of a finite collection of $\cdot$/M/1 nodes S_i, $i = 1, 2, ..., M$. The average service rate at each S_i is r_i. Let $\mathbf{N}_i^n$ be the service process of node i of network n (this means $\mathbf{N}_i^n$ is a rate r_i Poisson process, independent of all other processes). After being served at the i^{th} node, a customer is routed to node j with probability p_{ij}, where $\sum_j p_{ij} \leq 1$. The customer leaves the queueing system after being served at S_i with probability $\gamma_i = 1 - \sum_j p_{ij}$. At this point it is convenient to think of each $\mathbf{N}_i^n$ as decomposed into Poisson processes $\mathbf{N}_{i,j}^n$ and $\mathbf{N}_{i,o}^n$, where $\mathbf{N}_{i,j}^n$ and $\mathbf{N}_{i,o}^n$ are obtained by sampling $\mathbf{N}_i^n$ in a Bernoulli fashion according to the probabilities p_{ij} and γ_i respectively. The interpretation is that $\mathbf{N}_{i,j}^n$ and $\mathbf{N}_{i,o}^n$ represent the "potential transitions" of a customer from node i to node j or to the output provided queue i is non-empty prior to an epoch of the service process $\mathbf{N}_{i,j}^n$ or $\mathbf{N}_{i,o}^n$ respectively. We suppose that the rates r_i and the p_{ij} are such that the system is quasireversible if customers arrive according to a Poisson process of rate α and are assigned upon arrival at J_n to node S_i with probability ν_i. Each server in J_n is assumed to operate independently of all other servers in J_n and also of those in $J_m, m \neq n$. Therefore, we

are considering departures from a series of independent and identical Jackson Networks.

Theorem 1: *Given a series of independent and identical Jackson Networks, J_n, suppose the arrival process to the first network, $\mathbf{A}^1$, is stationary and ergodic of rate α. If $\mathbf{A}^n$ is the arrival process to the n^{th} network, and if α is small enough so that each network in the series is stable, then as n tends to infinity, $\mathbf{A}^n$ tends weakly to a mixed Poisson process of average rate α.*
Outline of the proof: Couple $\mathbf{A}^1$ with an independent, rate α, Poisson process $\mathbf{P}^1$ and define the process of yellow, red and blue customers $\mathbf{Y}^n$, $\mathbf{R}^n$ and $\mathbf{B}^n$ as before. Then each $\mathbf{P}^n$ is Poisson and $I(\mathbf{Y}^n)$ which is equal to the average rate of the yellow customers at stage n is increasing. Let $\gamma_i > 0$ for i belonging to a certain subset I of $\{1, 2, ..., M\}$. Let $\mathbf{D}^n$ be the "potential departure process" at the n^{th} stage. $\mathbf{D}^n$ is the superposition of the Poisson processes $\mathbf{N}^n_{i,o}, i \in I$ and therefore is a Poisson process of rate $\sum_{i \in I} r_i \gamma_i$ and *each point of the process $\mathbf{A}^n$ is a point of $\mathbf{D}^n$* (likewise every point of $\mathbf{P}^n$ is a point of $\mathbf{D}^n$). From this we deduce that the joint process $(\mathbf{A}^n, \mathbf{P}^n)$ is tight and therefore there exist weak subsequential limits of this sequence. It is first shown that all the subsequential limits are the same and equal to $(\mathbf{A}^\infty, \mathbf{P}^\infty)$, say. Then we show that the joint process $(\mathbf{A}^\infty, \mathbf{P}^\infty)$ is invariant to the operation of passing through a network like J_n and therefore argue that $\mathbf{A}^\infty$ has to be mixture of Poisson processes. The proof technique is similar to that used in Section 3, Chapter VIII of Liggett [7]. ■

The reason that we may not have a Poisson limit in Theorem 1 is that blue and red customers may end up overtaking each other by traversing different paths of a network and therefore avoiding coalescing. The next theorem shows that if certain topological and connectivity conditions are imposed on the networks J_n, we can show that the $\mathbf{A}^n$ converge to a Poisson process.

Theorem 2: *Suppose the sequence of networks J_n of Theorem 1 satisfy the following conditions (a) $\min_i \nu_i > 0$, (b) there exists node S_i such that $\min_j p_{ji} > 0$ and $\min_j p_{ij} > 0$, then $\mathbf{A}^n$ converges to a rate α Poisson process in distribution.* ■

With the aid of the following examples, we explore the intuition behind Theorems 1 and 2 and discuss their implications. Consider the networks of Figure 2.

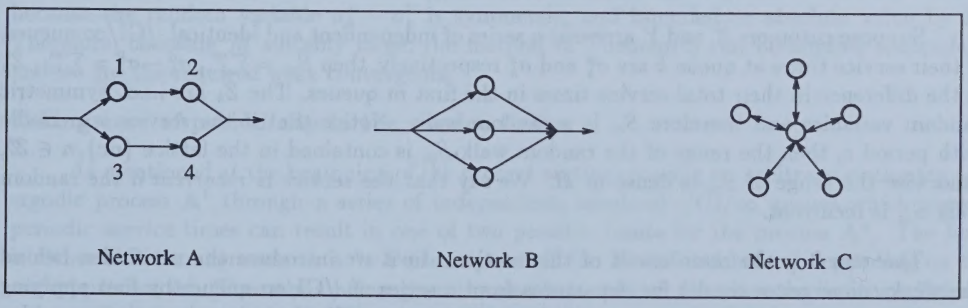


Figure 2: Examples

Theorem 1 applies to all three types of networks. Therefore, in all three cases we have

convergence of successive departures to a mixed Poisson process. In addition, Network B satisfies condition (a) of Theorem 2, and Network C satisfies condition (b) of Theorem 2 with the node in the center acting as the "star point". Hence, for a series of networks of type B or type C, we get the desired Poisson convergence. Intuitively speaking, what conditions (a) and (b) of Theorem 2 ensure is this. Although a given red customer may overtake blue customers (or vice versa), this can happen only finitely often because there is a definite probability (bounded away from 0) that the red will coalesce with the blue that she is overtaking. That is, overtaking is allowed but at a price. It is precisely this that is not guaranteed in the case of Network A. Therefore, for departures from an infinite series of networks of type A, it is not yet known if the limit is a Poisson process.

3. Limiting Departures from a Series of $\cdot/GI/\infty$ Queues

It is known that the equilibrium departure process from a $\cdot/GI/\infty$ queueing system, when fed by a Poisson process, is a Poisson process ([11]). Vere-Jones [11] uses the method of probability generating functionals to show that if $\mathbf{A}^1$ is a stationary and ergodic *weakly mixing* point process and the J_n's (of Figure 1) are independent and identical copies of a $\cdot/GI/\infty$ node, then the processes $\mathbf{A}^n$ converge in distribution to a Poisson process. It is interesting to note that the result is false if the weak mixing condition is dropped. For, suppose that the deterministic process with unit inter-arrival times is fed to an infinite series of independent $\cdot/GI/\infty$ queues. Also suppose that the service times at these nodes are i.i.d. with equal probability of being 1 or 2 units. Then, pathwise, all the points of the departure process from J_1, $\mathbf{A}^2$, will be separated by integer distances (possibly zero). It then follows that the same is the case with each of the $\mathbf{A}^n$'s, making it impossible for the $\mathbf{A}^n$'s to converge to a Poisson process. In this section we use results from couplings for random walks to show that for *any* stationary and ergodic arrival process $\mathbf{A}^1$, the $\mathbf{A}^n$ converge in distribution either to a Poisson process or to a ν-Poisson process (defined below) depending on the type of service distribution. In particular, our result includes Vere-Jones' result as a special case.

Definition 1: We say that the service time of a customer is *periodic* with period c if the possible values of the service times are integer multiples of c. Otherwise the service is *aperiodic*.

Suppose cutomers X and Y arrive at a series of independent and identical $\cdot/GI/\infty$ queues. If their service times at queue k are σ_k^x and σ_k^y respectively, then $S_m = \sum_{k=1}^m \sigma_k^a - \sigma_k^p = \sum_{k=1}^m Z_k$ is the difference in their total service times in the first m queues. The Z_k are i.i.d. symmetric random variables and therefore S_m is a random walk. Notice that if the service is periodic with period c, then the range of the random walk S_m is contained in the lattice $\{nc\}, n \in \mathbb{Z}$, otherwise the range of S_m is dense in $\mathbb{R}$. We say that the service is *recurrent* if the random walk S_m is recurrent.

Theorem 3 is the main result of this section. In it we introduce the main idea behind the weak convergence results for departures from a series of $\cdot/GI/\infty$ queues by first applying it to the case where the service is aperiodic and recurrent. We then show how the same idea generalizes to other $\cdot/GI/\infty$ queues with appropriate modifications.

Theorem 3: *The limit from passing a stationary and ergodic arrival process of rate* α, $\mathbf{A}^1 =$

$\cdots\ t^a_{-1} < 0 \leq t^a_0 < t^a_1 \cdots$ *through an infinite series of independent, identical* $\cdot/GI/\infty$ *queues, whose service is aperiodic and recurrent is a Poisson process of rate* α.

Outline of proof: Couple $\mathbf{A}^1$ with an independent rate α Poisson process $\mathbf{P}^1$ by matching up points (or customers) of $\mathbf{A}^1$ and $\mathbf{P}^1$. As before, customers belonging to $\mathbf{A}^1$ are coloured blue and customers belonging to $\mathbf{P}^1$ are coloured red. A key observation is that so long as a red and a blue customer evolve independently of each other, they perform a random walk as regards to their inter-arrival times at each of the queues. Specifically, if customers C_A and C_P belonging to processes $\mathbf{A}^1$ and $\mathbf{P}^1$ respectively arrive at times t_a and t_p to the first queue, then $\tau^1 = t_a - t_p$ is the difference between their arrival times to the first queue. If σ^a_k and σ^p_k are their service times at the k^{th} queue, then the random variable

$$\tau^m = \tau^1 + \sum_{k=1}^{m-1} \sigma^a_k - \sigma^p_k = \tau^1 + \sum_{k=1}^{m-1} Z_k$$

is the difference in the arrival times of these customers at the m^{th} queue. Because of the recurrence and aperiodicity of the service, τ^m is a recurrent, aperiodic random walk on R with starting value τ^1. By our coupling scheme, $\tau^1 \leq R$ a.s. Thus, given $\epsilon > 0$, $|\tau^m| < \epsilon$ for some $m(\omega)$ a.s. Once they are sufficiently close, we say they are "coupled" (at which point they are coloured yellow) and let them evolve together forever. Thus, eventually, points of $\mathbf{A}^n$ and $\mathbf{P}^n$ are arbitrarily close to each other and we get the required weak convergence, since the $\mathbf{P}^n$'s are all Poisson. ■

If the service times σ^a_k and σ^p_k in the above theorem are such that the random walk $\tau^m = \tau^1 + \sum_{k=1}^{m-1} \sigma^a_k - \sigma^p_k$ is *transient* (but still aperiodic), then we modify the service as follows:

- For each k, either σ^x_k and σ^y_k are both bigger than M (to be specified later), or they are both smaller than or equal to M.
- If they are both bigger than M, then let $\sigma^x_k = \sigma^y_k$. If not, choose σ^x_k and σ^y_k to be less than or equal M *independently* of each other (and of the service times of the rest of customers).

In other words, big jumps are taken together and small jumps are taken independently. With this prescription, the random walk $\tau^m = \tau^1 + \sum_{k=1}^{m-1} \sigma^x_k - \sigma^y_k$ then becomes recurrent because the random variable $\sigma^x_k - \sigma^y_k$ is symmetric, and bounded in absolute value by M. Therefore, choosing M suitably large, the method of Theorem 3 can be applied successfully and we get the required weak convergence.

Taking care of periodicity in the service times

As mentioned at the beginning of the present section, passing an arbitrary stationary and ergodic process $\mathbf{A}^1$ through a series of independent, identical $\cdot/GI/\infty$ queues which possess periodic service times can result in one of two possible limits for the process $\mathbf{A}^n$. The limit is either a Poisson process or a "ν-Poisson process" (to be defined below) depending on the arrival process $\mathbf{A}^1$. Theorem 4 below gives the precise condition for when the limit is a Poisson process. But first we make the following definition.

Definition 2: Given $c > 0$, x (*mod* c) is the element of $[0,\ c)$, which taken away from x, leaves an integer multiple of c.

The following lemma is a prelude to the definition of a ν-Poisson process and leads to the statements of Theorems 4 and 5. The proof of Lemma 1 is based on the Ergodic Theorem.

Lemma 1 *Let* $\mathbf{A} = \cdots t_{-1} < 0 \leq t_0 < t_1 \cdots$ *be a stationary point process of finite intensity and* $\mathcal{P}^n(\omega) = \frac{1}{N^A([0,\ nc))} \sum_{t_i \in [0,\ nc)} \delta_{t_i\ (mod\ c)}$. *Then there exists a random probability measure* $\mathcal{P}^A(\omega)$ *on* $[0,\ c]$ *such that a.s.* $\mathcal{P}^n(\omega) \overset{weakly}{\longrightarrow} \mathcal{P}^A(\omega)$. ■

Definition 3: A point process $\mathbf{A} = \cdots t_{-1} < 0 \leq t_0 < t_1 \cdots$ is said to be *uniform on* $[0,\ c]$ if a.s.

$$\frac{1}{N^A([0,\ nc))} \sum_{t_i \in [0,\ nc)} \delta_{t_i\ (mod\ c)}$$

converges to the uniform distribution on $[0,\ c]$ as $n \to \infty$.

We now obtain the following strengthening of Theorem 3.

Theorem 4: *Let* $\mathbf{A}^1$ *be a stationary and ergodic arrival process of rate* α *and let* $\mathbf{A}^n$ *be the departure process from the* $(n-1)^{th}$ *queue when* $\mathbf{A}^1$ *is fed through an infinite series of independent, identical* $\cdot/GI/\infty$ *queues. Then* $\mathbf{A}^n$ *converges in distribution to a Poisson process of rate* α *if and only if*
(a) The service is aperiodic
or
(b) The process $\mathbf{A}^1$ *is uniform on* $[0,\ c]$, *the period of the service times.* ■

Given a probability measure ν on $\mathbb{R}$ with support in $[0,\ c]$, define the measures $\nu^n, n \in \mathbb{Z}$ with support in $[nc,\ (n+1)c]$ as translates of ν by nc units of time. That is $\nu^n(S) = \nu(S\ (mod\ nc))$ for a Borel sets S in $[nc,\ (n+1)c]$. Note that $\nu^0 = \nu$.

Definition 4: Given $c > 0$ and a measure ν on $[0,\ c]$, the process $\mathbf{P}_\nu$ is said to be a *ν-Poisson process of rate* α if the following conditions hold.
i) The number of points of $\mathbf{P}_\nu$ in $[nc,\ (n+1)c)$ is an i.i.d. sequence, as n varies over $\mathbb{Z}$, with each marginal being distributed as a Poisson parameter αc random variable.
ii) Each point of $\mathbf{P}_\nu$ in $[nc,\ (n+1)c]$ is distributed over the interval $[nc,\ (n+1)c]$ according to ν^n, independent of all other points.

Note that according to the above definition $\mathbf{P}_\nu$ is the usual Poisson process with parameter α if ν is the uniform distribution over $[0,\ c]$, and it is a batch process of i.i.d. Poisson random variables if ν is a point mass at $x \in [0,\ c]$. Hence $\mathbf{P}_\nu$ is, in general, not a time stationary process. Given a probability measure ν on $[0,\ c]$ and a subset S of $[0,\ c]$, define the family of measures $\{\nu_\circ t\}, t \in \mathbb{R}$ on $[0,\ c]$ as $(\nu_\circ t)(S) = \nu[(S+t)\ (mod\ c)]$. The $\nu_\circ t$ are translates $(mod\ c)$ of ν as t varies.

For a stationary and ergodic point process $\mathbf{A}$, the associated limiting empirical measure $\mathcal{P}^A(\omega)$ (as defined Lemma 1) can be written as $\mathcal{P}^A(\omega) = \nu_\circ T(\omega)$ where $T(\omega)$ is uniformly distributed on $[0,\ c]$. This is because for each ω, $\mathcal{P}^A(\omega)$ is a translate $(mod\ c)$ of some measure ν. By ergodicity this translation, say T, must then be uniform. This implies that $\mathcal{P}^A(\Theta_T(\omega)) =$

$\nu(\omega)$. Define the process $\mathbf{A}_\nu$ as $\mathbf{A}_\nu(\omega) = \mathbf{A}(\Theta_T(\omega))$ and say that $\mathbf{A}_\nu$ is the *ν- periodic translation of* $\mathbf{A}$. Clearly the emperical measure $\mathcal{P}^{A_\nu}$ associated with $\mathbf{A}_\nu$ is a.s. identically equal to the measure ν. Note that $\mathbf{A}(\omega) = \mathbf{A}_\nu(\Theta_{-T}(\omega))$, and that the process $\mathbf{A}_\nu$ is not Θ_t stationary in general (it will, however, be Θ_{nc} stationary). Given a ν-Poisson process $\mathbf{P}_\nu$, call the Θ_t stationary process $\mathbf{P}_{\nu,S}$ defined by $\mathbf{P}_{\nu,S}(\omega) = \mathbf{P}_\nu(\Theta_{-T}(\omega))$ a *stationary ν-Poisson process.*

Theorem 5: *Let $\mathbf{A}^1$ be a stationary and ergodic process of rate α and suppose*

$$\lim_{n\to\infty} \frac{1}{N^{A^1}([0,\ nc))} \sum_{t_i \in [0,\ nc)} \delta_{t_i\ (mod\ c)}$$

converges a.s. to a translate (mod c) of some distribution ν on $[0, c]$. Let $\mathbf{A}^n$ be the departure process from the $(n-1)^{th}$ stage of an i.i.d. series of $\cdot/GI/\infty$ queues when $\mathbf{A}^1$ is inputted to the first queue. Suppose the service time of each customer is periodic with period c. Then $\mathbf{A}^n$ converges in distribution to a stationary ν-Poisson process of rate α. ■

Remarks

We refer to the tree network shown in Figure 3 below. Let the arrival process to the first node be an arbitrary stationary and ergodic process of rate α. Suppose it is split by Bernoulli sampling (no queueing operation is performed), into two processes each of rate $\alpha/2$. These processes are further split by Bernoulli sampling into processes of rate $\alpha/4$, and so on. It is a well-known result of point process theory that if time is rescaled suitably so that at each step the "baby processes" are all of rate α, then these baby processes are converging to a Poisson process in distribution.

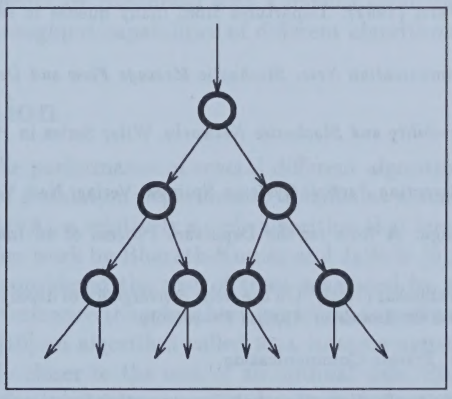


Figure 3: A Tree Network

Based on our results for convergence of departures from a series of $\cdot/GI/\infty$ queues, it is easily seen that if we at each step we first pass the arrival process through a $\cdot/GI/\infty$ queue and then split it by Bernoulli sampling, and rescale time suitably the baby processes again converge either to a Poisson process or to a ν-Poisson process depending on the original arrival process

and the service distribution. Now suppose that instead of passing the arrival process through a $\cdot/GI/\infty$ at each step and then splitting it, we pass it through a $\cdot/M/1$ node (or a $\cdot/M/k$ node). Then although the baby processes do converge to a mixed Poisson process (follows from Theorem 1), we cannot conclude that the limit is a pure Poisson process.

Likewise, in the case of Network A of Figure 2, if each of the nodes 1, 2, 3 or 4 was a $\cdot/GI/\infty$ node then again using the I-S coupling we can conclude that the limit from passing an arbitrary (stationary and ergodic) arrival process $\mathbf{A}^1$ through an infinite series of such networks results in a Poisson process (or a ν-Poisson process) limit.

As a final remark, it is worth pointing out that although our results have been obtained for departures from a series of identical networks, it is not hard to see that due to the nature of the Liggett Coupling our arguments hold in some situations where the networks are not identical.

References

[1] V. Anantharam (1993): Uniqueness of stationary ergodic fixed point for a $\cdot/M/k$ node. *Annals of Applied Probability*, **3**, 1, 154-173.

[2] N. Bambos and J. Walrand (1990): An Invariant Distribution for the G/G/1 Queueing Operator. *Advances in Applied Probability*, **22**, 254-256.

[3] M.Ekhaus and L.Gray (1993): A Strong Law for the Motion of Interfaces in Particle Systems. *In preparation.*

[4] P. W. Glynn and W. Whitt (1991): Departures from many queues in series. *The Annals of Applied Probability*, **1**, 4, 546-572.

[5] L. Kleinrock (1964): *Communication Nets: Stochastic Message Flow and Delay*, Dover Publications, New York.

[6] F. P. Kelly (1979): *Reversibility and Stochastic Networks*, Wiley Series in Probability and Mathematical Statistics.

[7] T. M. Liggett (1985): *Interacting Particle Systems*, Springer-Verlag, New York.

[8] T. M. Liggett and T. Shiga: A Note on the Departure Process of an Infinite Series of $\cdot/M/1$ Nodes. *Unpublished manuscript.*

[9] T. S. Mountford and B. Prabhakar (1993): On the weak convergence of departures from an infinite sequence of $\cdot/M/1$ queues. To appear in *Annals of Applied Probability.*

[10] M. Reiman and B. Simon: Private Communication.

[11] D. Vere-Jones (1968): Some applications of probability generating functionals to the study of input-output streams. *Journal of the Royal Statistical Society, Series B*, **30**, 321-333.

Shortest Path Multipoint Routing: experimental studies

Bernard M. Waxman *

Department of Computer Science
Southern Illinois University at Edwardsville
Edwardsville, IL 62026
bmw@mpr.siue.edu

Abstract

The use of shortest path routing from destination to source has been suggest for multipoint routing. The idea is to apply standard point-to-point routing between the root of a multipoint tree and a node that wishes to join the tree. This technique has both the advantage of being relatively simple and of building on all that is known about point-to-point routing. Several earlier experimental studies have shown that this technique generates trees of reasonable cost. In this paper, we present the results of experiments in which shortest path routing compares favorably to several more complex routing techniques. These experiments are conducted with a routing simulator that allows us gather information on the throughput capabilities of different algorithms.

1 Introduction

This paper explores the performance of several different algorithms for multipoint routing through the use of simulation experiments. Particular attention is directed toward *shortest path routing* (SPR), a relatively simple algorithm that lends itself to a distributed implementation. Earlier work by Bharath-Kumar and Jaffe in [5], Doar and Leslie in [3], and Waxman in [10] considered the cost of trees generated by SPR. In each case, SPR offered reasonable performance though other algorithms sometimes did significantly better. For example, in [10] an algorithm called RGA in some experiments generated trees with costs significantly closer to the cost of an optimal tree. Since the real measure of performance of a routing algorithm is not tree cost but throughput or blocking rates, the performance of SPR is compared to several different algorithms by measuring the throughput capabilities with a network simulator.

After considering several different approaches to modeling multipoint connections we have selected one on the basis of simplicity and generality. Since it is difficult to anticipate future network applications, it is difficult to choose a "good" multipoint model for experimental studies. Our choice of models is based in part on the assumption that any algorithm must support dynamic connections, a connection where the set of nodes in a connection is not static, and should also support connections that allow for more than one source. For example, an application such as a video conference is likely to be

*This work supported by NSF grant NCR-9210548.

dynamic with hosts joining and leaving the connection at different points in time, and it is likely that many hosts may take turns as the broadcast source.

Our connection model is based on a single spanning tree with bidirectional, symmetric bandwidth reservation. For some applications, it may be reasonable to reserve a different bandwidth for each direction, e.g., a television broadcast might need bandwidth in only one direction. The algorithms considered have the capability of dealing with connections that do not require bidirectional bandwidth. Still, we conduct experiments only for connections with bidirectional bandwidth allocation primarily because of the problem of approximating minimum cost trees.

2 Underlying Graph and the Dynamic Steiner Tree Problem

Our network model is an undirected graph where edges corresponds to network links and nodes correspond to switches. Each edge has a cost and a bandwidth or capacity. Since the capacity of each edge is assumed to be symmetric, an undirected graph is the appropriate model. The network itself is connection oriented in the style of ATM networks. When a connection is first established, a bandwidth requirement is specified. Only those links with sufficient free capacity to support a connection's bandwidth requirement may be used.

The *dynamic Steiner tree problem* [4] serves as the basis for our multipoint model. Briefly, an instance of this problem consists of an undirected graph with an edge metric and a sequence of node requests. Each request either asks us to add a node to or remove a node from a terminal set. The set of nodes to be connected is called the terminal set, and its elements are called terminal nodes. The problem is to find a sequence of minimum cost Steiner trees that span the terminal set. In [4], it was shown that deterministic non-rearrangeable algorithms for the on-line problem have a competitive ratio $\Omega(\log k)$ where k is the number of nodes in the terminal set.

When *remove* requests are allowed, no lower bound for worst-case performance exists [4]. Even though this result is not very promising, the average performance of non-rearrangeable algorithms can be quite good. When the Dynamic Steiner problem is modified to require that the owner remain in the connection, an algorithm such as *shortest path routing* (SPR) yields performance within k of optimum.

3 Connection Model and Performance Evaluation

The criteria used to measure performance is the ability of an algorithm to maximize throughput. By minimizing connection cost, we attempt to maximize the use of a network's resources. This also helps to keep communication delay to a minimum since the network will not be overloaded unnecessarily. What none of our algorithms do is to explicitly allow us to address quality of service issues. For example, for a connection where delay is critical, we might prefer an algorithm that takes this into account. Some algorithms can be modified to make use of such considerations by designing a cost function that uses quality of service as one of its parameters or by using a technique like that presented by Khuller et al in [6].

3.1 The connection model

For these experiments, a multipoint connection consists of a collection of hosts, a connection tree, and a bandwidth requirement. In order simplify the simulation, we assume that all switches are non-blocking. In a real network, even though switches may not be truly non-blocking, it is likely that they will have very low blocking rates.

In a connection, each host is identified with the terminal node to which it is connected. Each connection also contains a distinguished host, (terminal node) known as the *connection owner*. The owner remains during the entire lifetime of a connection while other nodes may come and go. Each connection evolves through a sequence of requests. All requests, except for the close-connection, involve one terminal node at a time, allowing for the greatest flexibility. The four types of requests are listed below.

Open-connection consists of a host, called the owner, and a bandwidth requirement.

Add-host consists of a host and the connection to which it is to be added.

Remove-host consists of a host and the connection from which it is to be deleted.

Close-connection consists of a connection that is to be terminated.

Since more than one host may be attached to any given terminal node, it is possible to have multiple add-host or remove-host requests for a single node. A remove-host request will never refer to the owner of a connection since the owner remains in the connection until a close-connection is issued.

3.2 Performance measurement

A primary goal is to find *practical* algorithms that maximize the load carried by a given network. For example, an algorithm that routes 95% of the connection requests is superior to one that routes only 80% of the requests. Performance is measured in terms of carried load, blocking rate, and load blocking rate as functions of offered load. Several researchers have investigated a theoretical version of this problem for point-to-point routing, e.g., Awerbuch, Azar, and Plotkin in [1].

Since add-host is the only request that can block, the blocking rate is A_B/A, where A_B is the number of requests blocked and A is the total number of add-host requests. A remove-host or a close-connection request can always be serviced since both free resources, and an open-connection request does not require any resources. A routing algorithm will service a request to add host h to connection c if it can find a path from h to c with sufficient free capacity to meet the bandwidth requirements of c.

We define the offered load to be the product of a connection's bandwidth requirement and the cost of a minimum Steiner tree for the connection's terminal set. This tree is computed without regard to link capacities. Thus, this measure removes any dependency on the current state of the network or the routing algorithm. Mathematically, the offered load O is defined as

$$O = \sum_{c_i \in \mathcal{S}} \text{cost}(T_i) B_i$$

where $\mathcal{S}$ is the set of all connections, $\text{cost}(T_i)$ is the cost of a minimum Steiner tree for all terminal nodes in connection c_i, and B_i is the bandwidth requirement for c_i. Since calculating O in this manner is an intractable problem, it is necessary to use an approximation algorithm to calculate each tree T_i.

Carried load C is defined similarly, except that T_i is replaced by a minimum tree for the terminal nodes that have not blocked. Thus, $C < O$ if any request to add a node to a connection has blocked. The load blocking rate L_b is just the fraction of offered load that blocks, or

$$L_b = 1 - \frac{C}{O} .$$

The cost of near-optimal Steiner trees are determined using IMST, a variation of the minimum spanning tree heuristic (MST) [5]. Even though both IMST and MST could serve as an algorithms for multipoint routing, neither is a reasonable choice for real world network applications due to their computational complexity and their need for global network information.

3.3 The simulation tool and load generation

Our model requires that a connection tree be established among a set of terminal nodes. All packets sent by members of a connection travel along the links of this tree until the connection terminates. Since the set of terminal nodes may change over time, this tree is not necessarily fixed but may change over time.

Edge cost is a unitless integer value and capacity is an integer value that specifies the bandwidth in kilobits per second. Each node corresponds to a network switch that is connected to other switches by graph edges. In addition, each switch may have hosts connected to it. These hosts are not explicitly part of the graph, but instead an integer value, called *frequency*, is associated with each node. When a host (computer, telephone, etc.) is added to a connection, the simulator chooses a node to be identified with that host. The frequency associated with each node determines the probability that a new host will be identified with it. Given the set of nodes V, a function $f : V \rightarrow \mathcal{N}$ gives the frequency for each node $v \in V$. The probability $P(v)$ that a node v will be selected for a new host is given by

$$P(v) = \frac{f(v)}{\sum_{u \in V} f(u)}.$$

This allows us to model different traffic loads at each node. If a node has frequency 0, then no host may connect to it; otherwise a node may have many hosts with the maximum number limited only by the available bandwidth.

All network parameters are static in order to simplify the simulation. We concentrate on the actual routing strategy, assuming that there is a system in place which maintains the routing tables. Extending the simulator to allow for a dynamic routing tables due to conditions such as link failures or to model dynamic edge cost, though not part of the current work, should be considered in the future. Additional provisions to handle quality of service and network failures can be added later as they need not have a significant impact on the basic routing strategy.

In order to generate a load on a network, a simple probability distribution is used to determine whether a requested is to add or remove a node from a given connection. The probability that a request r is an add request is given by

$$P_C(k) = \frac{\gamma(n-k)}{\gamma(n-k) + (1-\gamma)k}$$

where n is the number of nodes in the network. The value γn equals the number of hosts in a connection at equilibrium [8]. The frequency of open-connection requests is used to control the load.

4 Multipoint Routing Algorithms

In addition to SPR, three classes of routing algorithms are considered. These are the *weighted greedy algorithm* (WGA), the *rearrangeable greedy algorithm* (RGA), and a version of the *minimum spanning tree heuristic* (IMST). In fact, SPR is a specific version of WGA that creates a shortest path tree rooted at a connection's owner. With SPR, each add-host is serviced by finding a shortest path to the connection owner that has sufficient free capacity. The connection tree is extended by using only that part of the path needed to connect the new node. When a remove-host request is received, any path in the tree that is no longer needed is removed. In some cases, the connection tree does not change, e.g., if the terminal node associated with the host is not a leaf node.

4.1 Weighted greedy algorithm

WGA is a collection of algorithms that generalizes SPR. These algorithms depend on a parameter $\omega \in [0, 0.5]$. So for example, we will refer to the algorithm with $\omega = 0.3$ as WGA(0.3). WGA handles a remove-host requests as does SPR. To service an add-host request, WGA joins a new node through a path to the existing connection tree $T = \{V_T, E_T\}$. To add a node u to a connection, a node $v \in V_T$ is selected to minimize the function

$$W(u,v) = (1-\omega)d(u,v) + \omega d(v,v_0) . \quad (1)$$

where $d(x,y)$ is the distance between nodes x and y. The parameter ω determines the weight given to the distance from u to v relative to the weight given to the distance from v to the connection-owner v_0. If $\omega = 0$ only the distance from u to v is considered. Thus, a shortest path to the connection is selected. When $\omega = 0.5$, the distance from u to v and the distance from v to v_0 are given equal weight. This is SPR since a shortest path to the connection owner is selected. Figure 1 gives pseudocode for WGA, where S is the set of terminal nodes in tree T.

4.2 Rearrangeable greedy algorithm

The *rearrangeable greedy algorithm* (RGA) is a collection of algorithms that is based on a straight forward enhancement to WGA. This algorithm limits the number of rearrangements to one path modifications per request. In those cases where the modification does not reduce the tree cost, no modification is made. As with WGA, this algorithm depends on the parameter $\omega \in [0, 0.5]$.

Informally, RGA handles a remove-host request by pruning that portion of the connection tree which is no longer needed. If no branch can be pruned, the algorithm will attempt to reroute a single path in the connection provided that severing the node identified with the remove request disconnects just a single terminal node. Then the path to that terminal node is removed and a new path is selected to rejoin the node to the connection. The node is reconnected as a new add-request (without rearrangement). This step helps to eliminate unnecessarily long paths to single terminal nodes.

```
T := {{v0}, ∅}
S := {v0}
do i ∈ {1,2,...,k} →
    If request = add(u) →
        select v for which min_{v∈V_T} W(u,v) is attained
        let p(u,v) be a shortest path from u to v
        T := T ∪ p(u,v)
        S := S ∪ {u}
    | request = remove(u) →
        S := S − {u}
        do V_T − S contains a node w with degree 1 →
            T := T − w      //remove w and the edge incident with w
        od
    fi
od
```

Figure 1: The weighted greedy algorithm WGA

To service an add-host request, RGA first joins a new node u through a path to the existing connection tree $T = \{V_T, E_T\}$ just as WGA would. Then it selects a second path, a shortest path from the new node to a node in V_T that is not in the path from u. The cycle created is eliminated by removing the longest path that still leaves a connected tree (primitive path). Note that it is possible for the primitive path to be identical to the second path. When that occurs, the rearrangement step has no effect on the resulting tree. For pseudocode, see [10].

4.3 Finding a path with sufficient bandwidth

The actual implementation of WGA and RGA is complicated by the fact that any path selected must have sufficient free capacity to meet the needs of the connection. In our simulation, information about available link capacity is logically distributed throughout the network at each node. A path is selected by searching along the shortest path from the new endpoint u to the node v in the existing connection as selected by the routing algorithm. When a link is found to have insufficient free capacity alternative links are selected. Routing tables at each node contain a list of alternate links for a given destination, ordered by increasing path cost. A parameter λ is used to impose a limit on the search for alternate paths. An attempt to satisfy an add request is considered blocked after $\lambda + 1$ edges in the search are found not useable. This method of searching for a feasible path was selected on the grounds that there is a simple distributed implementation and good results can be obtained with small values of λ . Other less computationally efficient methods might be expected to find paths of somewhat lower cost.

4.4 MST for multipoint routing

We also make use of IMST as a routing algorithm in our experiments. This algorithm requires global information and allows complete rearrangement of the connection after each request. Therefore, it is not practical as a multipoint algorithm. On the other hand,

its ability to find near minimum cost trees makes it useful as a yardstick against which to compare the performance of other algorithms.

5 Performance

To get a preliminary comparison among the algorithms, we collected data from experiments that measure the cost of trees constructed on both random networks and networks that have been designed by hand. These experiments compared the cost of trees generated by WGA and RGA to near-optimal solutions generated by IMST. Then we investigated the more complex issue of throughput with a network simulator.

In previous work, it has been shown that simple algorithms, such as SPR, yield relatively good performance. Experiments that compare the cost of near-optimal solutions to the costs of solutions generated by SPR (WGA(0.5)) generally resulted in cost ratios within 25% to 50% of optimal [3, 8]. Using a weight of 0.0 (greedy-routing) sometimes does better, particularly when requests only add terminal nodes.

5.1 Connection costs

In experiments with a single connection, we discovered a significant variations in tree costs depending on the algorithm used. In general, RGA generates trees with lower cost than does WGA. For example, in one set of experiments the trees generated by SPR have a cost ratio of 1.30 while RGA(0.5) generates trees with a cost ratio of only 1.18. In the same set of experiments, RGA(0.0) produced a ratio of just 1.06 while WGA(0.0) produced a ratio of 1.14. Therefore, in this set of experiments the weight factor has an effect as significant as the rerouting of RGA. Not all experiments showed such large differences. Additional details on these and other experiments can be found in [10]. Figure 2 shows relative tree costs for SPR, IMST, and RGA on experiments conducted with the network simulator. The plots show the cost ratios averaged over all connection trees for a given offered load. In the case of RGA, plots for $\omega = 0.2$ and $\omega = 0.3$ are shown. In these experiments, the use of WGA(0.3) resulted in tree cost close to those generated by RGA. While the value of ω has very little effect on the cost of the trees generated by RGA.

5.2 Load blocking rates

When load blocking is evaluated with the simulator, the differences in tree costs does not relate very closely to throughput. In fact, the performance of SPR is very competitive with the more complex algorithms when load blocking and throughput are considered even though it generates trees with somewhat great cost ratios than RGA.

The experimental results presented in this section involve generating a load on a network by a collection of connections with different characteristics including both multipoint and point-to-point connections. See [9] for a description of the connections' characteristics. These experiments were conducted using three different networks. Network RAND is a random network containing 60 nodes and 106 edges, network CITY is a 56 node network with 99 edges organized into eight clusters, and network HIER is an hierarchical network consisting of 96 nodes and 105 edges arranged in a tree like structure with additional edges connecting local clusters. Both CITY and HIER were designed by hand with regular interconnection patterns in order to create a somewhat realistic network.

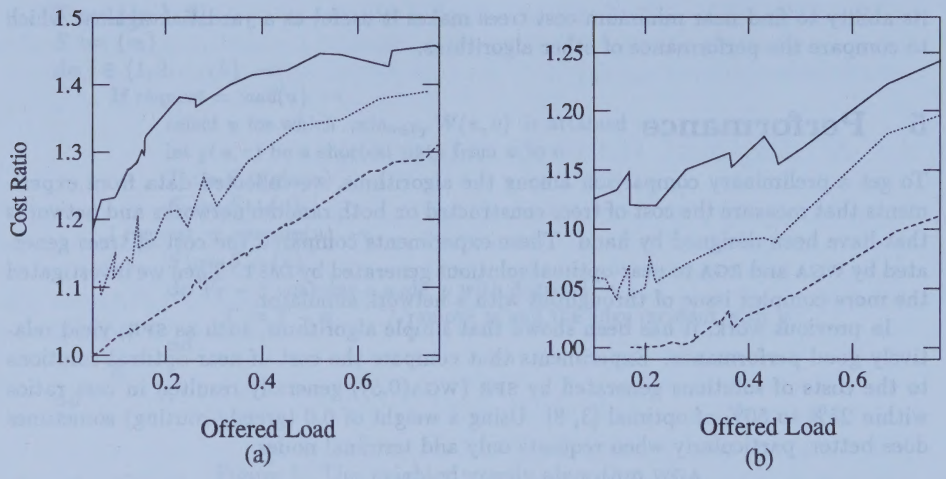


Figure 2: Average cost ratios for SPR (solid), RGA (dotted), and IMST (dashed). Graph (a) shows results for network RAND with RGA(0.3) and (b) for network CITY with RGA(0.2).

The parameter λ determines the number of alternate edges to consider in searching for a feasible path. A value in the range of 4 to 6 has given consistently good results. Smaller values of λ tend to degrade an algorithm's performance significantly while larger values generally result in only small gains. All experiments shown here use λ equal to 4.

Representative results are shown in the two figures below. Figure 3 illustrates the performance of SPR on networks CITY and RAND. The solid line plots the results for SPR and the dashed line plots IMST. In Figure 4 the dotted line is used for RGA(0.5). The performance of SPR does not vary significantly from WGA or RGA no matter what value of ω is used. Increasing λ does gives some improvement. With exhaustive search for a path, in most cases the improvement in performance is small. In the best-case, the distance between the curves for IMST and SPR is cut roughly by half. The relationship between SPR and IMST in graph (a) at offered loads less than 0.3 is surprising. Using exhaustive search instead of setting $\lambda = 4$ brings the curves only a little closer together. Part of this may be due to the nature of the experiments which results in larger statistical variations in offered load with a small average load. Undoubtedly some of this effect is due to the topology of network CITY since differences are smaller with the other networks.

It is somewhat surprising that the performance of the rearrangeable algorithm RGA is only slightly better than that of SPR. Certainly the improvement is not sufficient to justify the additional complexity. The use of values of *omega* other than 0.5 essentially has no significant effect. Thus, as is the case with RGA, the additional complexity of using $\omega \neq 0.5$ cannot be justified. Only when a large percentage of the load is due to connections that consist primarily of a sequence of add-host requests does the value of ω make any real difference. Then, some of the experiments show a small improvement in performance with $\omega = 0.0$. The only factor that produces a significant improvement in all experiments is the use of large values of λ. Even here the improvements are relatively small.

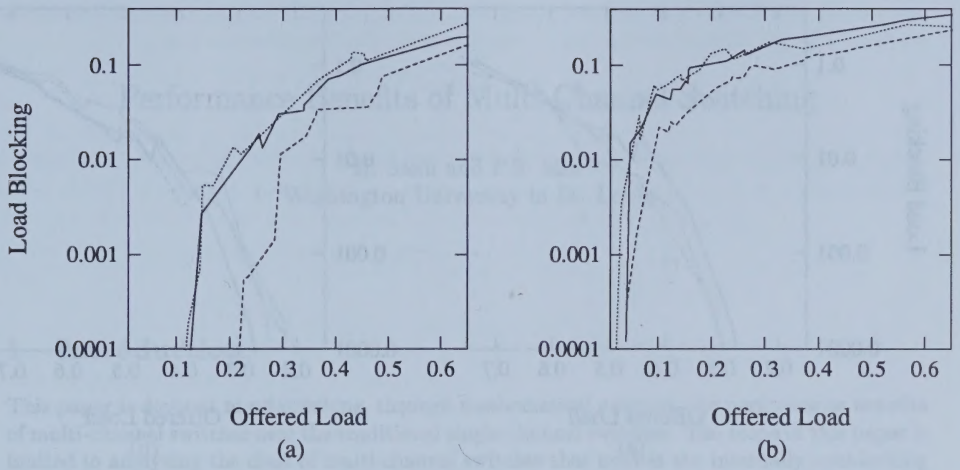


Figure 3: Load blocking rates for SPR (solid), RGA (dotted), and IMST (dashed). Graph (a) shows results network CITY and (b) for network RAND.

6 Conclusions and Future Research

Shortest path routing has given uniformly good results in all experiments conducted to date. Even though the more complex algorithms have been able to give a small reduction in blocking in some experiments, the improvements have not been sufficient to justify the significant increase in complexity. Under special circumstances, e.g., connections where the majority of requests are add-host, we do see more significant differences. In most cases, the performance of SPR is so close to the other algorithms that the fact that it is relatively simple and is based on point-to-point routing is most important.

An underlying principal in these experiments has been to minimize connection cost under the assumption that edge costs are fixed. None of the algorithm considered in this paper will block a request if it is able to find a feasible route. In order to obtain a competitive on-line algorithm, the algorithm in [1], sometimes blocks a request even when sufficient bandwidth is available. Furthermore, with this scheme, edge costs are not fixed, but are a function of the load they carry. The question remains open if using this type of algorithm will in practice result in improved throughput. Such a scheme presents new problems to a distributed network. Questions related to maintaining dynamic routing tables and the determination of reasonable and fair criteria for rejecting a request to save capacity for future requests need to be addressed. These questions present interesting possibilities for continued research and future experiments.

In summary, our experimental results present a strong case for shortest path routing. The simplicity of SPR along with its performance in comparison to more complex algorithms suggest that SPR or some variant of SPR may be the most reasonable choice for multipoint applications.

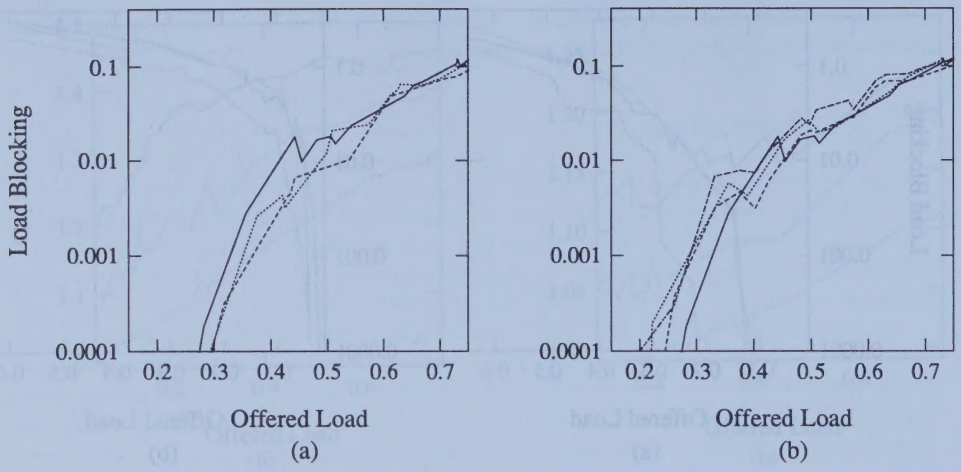


Figure 4: Load blocking rates for SPR (solid) on network HIER. Graph (a) shows results for RGA(0.5) dotted and IMST(dashed). Graph (b) shows results for WGA(0.3) (dotted), RGA(3) (dashed), and RGA(0.0) (dashed, dotted).

References

[1] B. Awerbuch, Y. Azar and S. Plotkin. "Throughput-competitive on-line routing." *Proc of 34th IEEE Symposium on Foundations of Computer Science.* October 1993, pp. 32-40.

[2] T. Ballardie, P. Francis and J. Crowcroft. "Core based trees (CBT)." *Proc of ACM SIGCOMM'93.* 1993, pp. 85-95.

[3] M. Doar and I. Leslie. "How bad is Naive multicast routing?" *Proc of IEEE INFOCOM'93.* 1993, pp. 82-89.

[4] M. Imase and B.M. Waxman. "Dynamic Steiner tree problem." *SIAM J. on Disc. Math.*, 3(1991), pp. 369-384.

[5] K. Bharath-Kumar and J.M.Jaffe. "Routing to multiple destinations in computer networks." *IEEE Trans.*, COM-31(1983), pp. 343-351.

[6] S. Khuller, B Raghavachari and N. Young. "Balancing minimum spanning and shortest path trees." *Proc of the Third Annual ACM-SIAM Symposium on Discrete Algorithms.* Jan 1993, pp. 1-8.

[7] D.C. Verma and P.M. Gopal. "Routing reserved bandwidth multi-point connections." *Proc of ACM SIGCOMM'93.* 1993, pp. 96-105.

[8] B.M. Waxman, *Evaluation of algorithms for multipoint routing*, Doctoral thesis, Washington University St. Louis, 1989.

[9] B.M. Waxman "Performance evaluation of multipoint routing algorithms." *Proceedings of IEEE INFOCOM'93.* 1993, pp. 980-986.

[10] B.M. Waxman "A rearrangeable algorithm for multipoint routing." *Proceedings of Thirty-First Annual Allerton Conference.* 1993, pp. 395-403.

Performance Benefits of Multi-Channel Switching

H. Saidi and P.S. Min
Washington University in St. Louis

1 Introduction

This paper is devoted to establishing, through mathematical analysis, the performance benefits of multi-channel switches over the traditional single-channel switches. The scope of this paper is limited to analyzing the class of multi-channel switches that exhibit the internally nonblocking property.

We assume that packets are of a fixed length. Due to the stochastic nature in the arrival patterns, some packets arriving in a time slot may not find enough output channels to exit from. It is thus necessary to delay transmitting of those packets thereby requiring buffers to store them until a later time slot. In [1] and [2], the authors show that the choice of buffering scheme affects the performance of internally nonblocking, single-channel switching networks significantly, and we anticipate similar effects in the context of multi-channel switching.

The goal of this paper is to verify the performance improvement due to channel grouping for various types of buffering schemes reported in the literature. To begin the analysis, we illustrated the performance gain in an internally nonblocking, multi-channel switching network that employs no buffer. Then we follow the outlines in [1] and [2], and analyze the multi-channel switching networks with *input buffering, shared buffering, and output buffering.*

To ease the notational burden and to make the identification of issues clearer in what follows, we assume independent and uniform arrival for packets: Arrival process at different input channels are independent, and a packet appearing at an input channel is equally likely destined for all channel groups at the output.

Suppose a switching network is given with N input channels and N output channels which are labeled between 0 and $N-1$. The output channels are divided into t equal sized channel groups, each with $m = \frac{N}{t}$ channels where t is a positive integer resulting in an integer value for m. We label the t channel groups between 0 and $t-1$ for convenience. Let p be the probability that there is a packet in a time slot at each input channel, and hence $\frac{p}{t}$ is the probability that there is an arriving packet destined for a specific channel group.

2 Bufferless Networks

Suppose that the given switching network has no buffer. The arrived packets at the input are lost if they cannot find an idle output channel belonging to the destined channel group to exit from in the same time slot. Let A_n^i be the random variable corresponding to the number of packets arriving in time slot n that are destined for channel group i. Suppose $A_n^i = k$. Since

there are m channels for each channel group, if $k > m$, the excess number, $k-m$, of the packets are lost. The probability of observing k arrived packets for channel group i is,

$$Pr(A_n^i = k) = \binom{N}{k} \left(\frac{mp}{N}\right)^k \left(1 - \frac{mp}{N}\right)^{N-k}.$$

We let L to be the probability that a packet destined for the considered channel group is lost. Then,

$$\begin{aligned}
L &= \frac{1}{mp} \sum_{k=m+1}^{N} (k-m) \binom{N}{k} \left(\frac{mp}{N}\right)^k \left(1 - \frac{mp}{N}\right)^{N-k} \qquad (1)\\
&= \frac{1}{mp} \sum_{k=0}^{N} (k-m) \binom{N}{k} \left(\frac{mp}{N}\right)^k \left(1 - \frac{mp}{N}\right)^{N-k}\\
&\quad - \frac{1}{mp} \sum_{k=0}^{m} (k-m) \binom{N}{k} \left(\frac{mp}{N}\right)^k \left(1 - \frac{mp}{N}\right)^{N-k}\\
&= \frac{1}{mp} \sum_{k=0}^{N} k \binom{N}{k} \left(\frac{mp}{N}\right)^k \left(1 - \frac{mp}{N}\right)^{N-k}\\
&\quad - \frac{1}{mp} \sum_{k=0}^{N} m \binom{N}{k} \left(\frac{mp}{N}\right)^k \left(1 - \frac{mp}{N}\right)^{N-k}\\
&\quad - \frac{1}{mp} \sum_{k=0}^{m} (k-m) \binom{N}{k} \left(\frac{mp}{N}\right)^k \left(1 - \frac{mp}{N}\right)^{N-k}\\
&= \frac{1}{mp} \cdot mp - \frac{1}{mp} \cdot m - \frac{1}{mp} \sum_{k=0}^{m} (k-m) \binom{N}{k} \left(\frac{mp}{N}\right)^k \left(1 - \frac{mp}{N}\right)^{N-k}\\
&= 1 - \frac{1}{p} - \frac{1}{mp} \sum_{k=0}^{m} (k-m) \binom{N}{k} \left(\frac{mp}{N}\right)^k \left(1 - \frac{mp}{N}\right)^{N-k}.
\end{aligned}$$

If $m \ll N$, (1) can be approximated using $(1+x)^y \simeq e^{xy}$ for small values of x and $\frac{1}{y}$, and it becomes,

$$\begin{aligned}
L &\simeq 1 - \frac{1}{p} - \frac{1}{mp} \sum_{k=0}^{m} (k-m) \binom{N}{k} \left(\frac{mp}{N}\right)^k \left(1 - \frac{mp}{N}\right)^{N}\\
&\simeq 1 - \frac{1}{p} - \frac{e^{-mp}}{mp} \sum_{k=0}^{m} (k-m) \binom{N}{k} \left(\frac{mp}{N}\right)^k\\
&= 1 - \frac{1}{p} - \frac{e^{-mp}}{mp} \sum_{k=0}^{m} (k-m) \frac{N!}{k!(N-k)!} \left(\frac{mp}{N}\right)^k\\
&= 1 - \frac{1}{p} - \frac{e^{-mp}}{mp} \sum_{k=0}^{m} (k-m) \frac{(N-k)! \cdot N \cdot (N-1) \cdot (N-2) \cdots (N-k+1)}{k!(N-k)!} \left(\frac{mp}{N}\right)^k\\
&\simeq 1 - \frac{1}{p} - \frac{e^{-mp}}{mp} \sum_{k=0}^{m} (k-m) \frac{(N-k)! \cdot N^k}{k!(N-k)!} \left(\frac{mp}{N}\right)^k\\
&= 1 - \frac{1}{p} - \frac{e^{-mp}}{mp} \sum_{k=0}^{m} (m-k) \frac{(mp)^k}{k!}. \qquad (2)
\end{aligned}$$

(2) has the maximum with $p = 1.0$, and it becomes

$$L \simeq e^{-m} \frac{m^m}{m!}. \tag{3}$$

The throughput, ρ, at each output channel is equals to $p \cdot (1 - L)$. Figure 1 shows the throughput with $N = 64$ as a function of p for various values of m. It is clear that with the size of the switching network fixed, the throughput is monotonically increasing with the increasing values of m, which confirm the suggested benefits of multi-channel switching.

For $m = 1$ (i.e., single-channel switching),

$$Pr(A_n^i = k) = \binom{N}{k} \left(\frac{p}{N}\right)^k \left(1 - \frac{p}{N}\right)^{N-k},$$

which becomes Poisson as N approaches ∞ such that

$$Pr(A_n^i = k) = \frac{p^k e^{-p}}{k!}.$$

Therefore the throughput, which equals the utilization of this output channel, becomes,

$$\rho = 1 - e^{-p},$$

and the maximum value of the throughput occurs at $p = 1.0$ where $\rho = 1 - e^{-1} = 0.63$.

3 Input Buffering

The input buffering scheme employs a buffer module at every input channel containing a certain number of buffers. Each of the buffers is able to store a single packet. A packet arrives at input channel at the beginning of a time slot, and waits in the buffer module moving forward with each packet from this buffer module that passes through the switching network until it moves to the head of the buffer. We assume those packets arriving to empty buffer modules are immediately available for routing through in the same time slot; this feature of the scheme is referred to as *cut-through* and is reported in the literature to improve the performance.

In order to assert the performance benefits for input buffering by channel grouping, we evaluate the throughput for a nonblocking switching network that employs buffer modules of infinite size. Thus newly arrived packets can always find an empty position in the buffer module and no packets are lost. This approach is consistent with the ones in [1] and [2], and we aim to derive the maximum throughput of such a switching network without consideration of specific buffer sizes.

Packets are served in a first-in-first-out (FIFO) manner from the buffer modules. With input buffering, the packets at the head of an buffer module, which includes those that arrive to an empty buffer module, can compete for output channels; thus at most N packets can be transmitted to the output in a time slot.

Among the packets at the head of the N buffer modules, if there are more than m packets destined for any channel group, the excess packets must remain in the buffer until the next time slot when they can contend once again for available output channels in the destined channel group.

Due to the FIFO service policy, the performance of the input buffering scheme can suffer from the phenomenon known as the head-of-line (HOL) blocking which limits the throughput of nonblocking single-channel switch to 0.58 as reported in [1] [2]. The HOL blocking prevents any packet not at the head of a buffer module from routing through the network, even if there is an idle output channel for the packet. We note that for the values of ρ near 0.58, the delay experienced by the arrived packets is very large. This means among other things that the buffer modules are likely to be occupied by many packets waiting to be transmitted to the output, and thus the probability that any buffer module is empty is small in each time slot.

Assume that each buffer module is saturated in the steady state such that there is at least one packet in it at all time. In time slot n, let q_n^i be the number of buffer modules with the HOL packet destined for channel group i which is blocked in time slot $n-1$ due to output contention. $\sum_{i=0}^{t-1} q_n^i$ is the number of buffer modules with HOL blocking in time slot $n-1$. F_n, which denotes the number of packets that move to the head of a buffer module in the beginning of time slot n, becomes,

$$F_n = N - \sum_{i=0}^{t-1} q_n^i.$$

F_n is also the number of packets that are routed through in time slot $n-1$.

Let A_n^i be the number of packets, amongst the F_n packets that freshly assumed the head position in a buffer module, that are destined for channel group i. Then,

$$Pr(A_n^i = k) = \binom{F_n}{k} \left(\frac{m}{N}\right)^k \left(1 - \frac{m}{N}\right)^{F_n - k}. \tag{4}$$

The throughput, ρ, equals the probability that a packet departs from a certain buffer module and routes through the network in a time slot, and from the symmetry among all channel groups,

$$\rho = \frac{F}{N} = 1 - \frac{1}{N}\sum_{i=0}^{t-1} q^i = 1 - \frac{q^i}{m}$$

where F and q^i are the average values for F_n and q_n^i respectively.

Let λ be the average number of packets that are at the head of a buffer module, that are destined for a certain channel group, and that are actually transmitted to the output. Again from the symmetry among the channel groups each consisting of m channels, $\lambda = m \cdot \rho$. Recalling that there is a packet in each of the N buffer modules and that these packets are equally likely destined among all channel groups, the average number of packets at the head that are destined for any channel group is m. Thus,

$$q^i = m - \lambda = m - m \cdot \rho. \tag{5}$$

Now considering for time slot n, the dynamics of each buffer module is described by,

$$q_{n+1}^i = max(q_n^i + A_n^i - m\ ,\ 0). \tag{6}$$

To determine the stationary distribution for q_n^i, we need to construct the balance equations governed by (6). Noting $Pr(A_n^i = j)$ is equal for all n and i due to the i.i.d. nature of the arrival process and the symmetry among the channel groups, we drop the indices n and i in what follows; A corresponds to the number of packets that freshly assume the head position

of a buffer module in the beginning of a time slot and that are destined for a certain channel group.

With p_j defined as $Pr(q_n^i = j)$,

$$\begin{aligned} p_0 &= p_0 \cdot Pr(A \le m) + p_1 \cdot Pr(A \le m-1) + \cdots + p_{m-1} \cdot Pr(A \le 1) + p_m \cdot Pr(A = 0) \\ &= \sum_{k=0}^{m} p_k \cdot Pr(A \le m-k) \\ p_j &= p_0 \cdot Pr(A = m+j) + p_1 \cdot Pr(A = m+j-1) + \cdots + p_{m+j-1} \cdot Pr(A = 1) \\ &\quad + p_{m+j} \cdot Pr(A = 0) \\ &= \sum_{k=0}^{m+j} p_k \cdot Pr(A = m+j-k) \quad \text{for } 1 \le j \le N-m. \end{aligned} \tag{7}$$

The z-transform for $p_j = Pr(q^i = j)$, denoted as $P(z)$, can be derived by multiplying both side of (7) by z^j and summing them over j.

$$\begin{aligned} P(z) &= \sum_{j=0}^{N-m} p_j \cdot z^j \\ &= z^0 \cdot \sum_{k=0}^{m} p_k \cdot Pr(A \le m-k) + \sum_{j=1}^{N-m} z^j \left(\sum_{k=0}^{m+j} p_k \cdot Pr(A = m+j-k) \right) \\ &= z^0 \cdot \sum_{k=0}^{m} p_k \cdot Pr(A < m-k) + \sum_{j=0}^{N-m} z^j \left(\sum_{k=0}^{m+j} p_k \cdot Pr(A = m+j-k) \right). \end{aligned} \tag{8}$$

After some algebraic manipulation, (8) becomes,

$$\begin{aligned} P(z) &= z^0 \cdot \sum_{k=0}^{m-1} p_k \cdot Pr(A < m-k) - \sum_{k=0}^{m-1} \sum_{j=0}^{m-1-k} z^{k+j} \cdot p_k \cdot Pr(A = j) \\ &\quad + z^{-m} \cdot A(z) \cdot P(z), \end{aligned} \tag{9}$$

where

$$A(z) = \sum_{j=0}^{N-m} Pr(A = j) \cdot z^j.$$

Solving (9) for $P(z)$ results in

$$P(z) = \frac{z^m \cdot \sum_{k=0}^{m-1} p_k \cdot Pr(A < m-k) - \sum_{k=0}^{m-1} \sum_{j=0}^{m-1-k} z^{k+j} \cdot p_k \cdot Pr(A = j)}{z^m - A(z)}. \tag{10}$$

The numerator of (10) is a polynomial of degree m, and thus has m complex roots. Since p_j is a probability, $P(z)$ is bounded and analytical on and inside the unit circle. Consequently, every root of the denominator in (10) on or inside the unit circle must be canceled by the numerator so they should also be the roots of the numerator. Thus (10) can be written as,

$$P(z) = \frac{K \cdot \Pi_{s=1}^{m}(z - z_s)}{z^m - A(z)} \tag{11}$$

where z_s is the s^{th} root of the denominator on or inside the unit circle and K is a certain constant to be determined shortly. Since $A(1) = \sum_{j=0}^{N-m} Pr(A = j) = 1$, the denominator vanishes at $z = 1$ and there is a root for the numerator at $z = 1$, which is labeled as the m^{th} root for convenience, i.e., $z_m = 1$.

To determine K, we use the fact that the limit of $P(z)$ as $z \to 1$ is one. Since $z = 1$ is the root of both the numerator and the denominator, we use the LeHopital's law, and,

$$lim_{z \to 1} P(z) = \frac{K(1 - z_1)(1 - z_2) \cdots (1 - z_{m-1})}{m - A'(z)_{|_{z=1}}} = 1. \tag{12}$$

Noting from (4) that A_n^i (and A) approaches Poisson for large values of N (and thus large values of F and F_n),

$$A(z) = e^{-\lambda(1-z)}, \tag{13}$$

and therefore,

$$K = \frac{m - \lambda}{(1 - z_1)(1 - z_2) \cdots (1 - z_{m-1})}. \tag{14}$$

Thus, once z_s is known for $s = 1, 2, \ldots, m - 1$, (14) determines K.

Using the properties of the z-transform we know that

$$q^i = \sum_{j=0}^{N-m} j \cdot p_j = \frac{d}{dz} P(z)_{|_{z=1}}.$$

Since both the numerator and the denominator of the derivative of $P(z)$ vanish at $z = 1$, the LeHopital's law needs be applied twice to $P(z)$ and as a result, q^i is determined as,

$$q^i = \sum_{s=0}^{m-1} \frac{1}{1 - z_s} - \frac{m(m-1) - \lambda^2}{2(m - \lambda)}. \tag{15}$$

From (5), we substitute for q^i with $m - \lambda$ in (15), and λ can be solved. The resulting value is valid if $\lambda < m$ for the stability of the buffer occupancy which is assumed implicitly. The throughput, ρ, then can be determined from $\lambda = m \cdot \rho$.

For $m = 1$, (16) simplifies to

$$1 - \lambda = \frac{\lambda^2}{2(1 - \lambda)}$$

resulting in $\lambda = 0.58$ which conforms the well known result of input buffering for single-channel switching.

When $m = 2$,

$$2 - \lambda = \frac{1}{1 - z_1} + \frac{\lambda^2 - 2}{2(2 - \lambda)}$$

where the z_1 is the only root of $z - e^{-\lambda(1-z)}$ inside the unit circle. Then, $\lambda = 1.372$ and $\rho = 0.686$. For the values of $m > 2$, the roots of $z^m - A(z)$ where $A(z)$ is given as (13) can be determined numerically. We used a numerical software package available at Washington University, namely the IMSL Library, and calculated the throughput, ρ, for different values of m by following the procedure outlined above. The results of such an exercise are shown in Figure 2. We see from Figure 2 the increasing values of ρ with respect to the increasing values of m, which asserts the benefits of channel grouping in multi-channel switching.

4 Shared Buffering

Shared buffering is achieved by providing a common buffer module that is accessible from all input channels. We assume again that the size of the common buffer module is infinite. The same assumption is employed in [1] and [2] where the performance of the shared buffering scheme for nonblocking single-channel switching is reported.

The arriving packets at different input channels join the buffer module in some arbitrary way. For an output channel group, packets are selected from the buffer module for routing to one of its m channels in a FIFO manner; each channel group receives up to m packets from the buffer module starting from the head position of the buffer module.

The maximum throughput for shared buffering can be seen to be 1.0 by the following argument: With the buffer module of infinite size, we note that the number of packets in the buffer module destined for different channel groups are independent. The i.i.d. nature of the arrival process implies that the average rate at which the number of packets arriving to the buffer module that are destined for a certain channel group equals $m \cdot p$. We claim that the departure rate for those packets is $m \cdot p$ for all $p \leq 1.0$. If the departure rate is less than $m \cdot p$, the number of packets in the buffer module grows large, which implies that there are more than m packets destined for the considered channel group. Then, m of such packets can be transmitted to the channel group, and the departure rate becomes m which is greater than or equal to the arrival rate $m \cdot p$. This argument states that the occupancy of the buffer module does not grow arbitrarily and that the departure rate must be equal to the arrival rate for all values of p. The maximum throughput thus becomes 1.0 regardless of the value of m.

In order to observe the benefits of multi-channel switching in shared buffering, we consider next the delay experienced by arrived packets. Let q_n^i be the number of packets in the buffer module that are destined for channel group i in the beginning of time slot n and let A_n^i be the number of newly arriving packets that are destined for channel group i. We recall that

$$Pr(A_n^i = k) = \binom{N}{k} \left(\frac{mp}{N}\right)^k \left(1 - \frac{mp}{N}\right)^{N-k}$$

which approaches Poisson as N gets large.

The dynamics of q_{n+1}^i is expressed as,

$$q_{n+1}^i = max(q_n^i + A_n^i - m\ ,\ 0). \tag{16}$$

We note that while the meanings of q_n^i and A_n^i are different, (16) is identical to (6), the balance equations determining the stationary distribution for q_n^i becomes,

$$\begin{aligned} p_0 &= p_0 \cdot Pr(A \leq m) + p_1 \cdot Pr(A \leq m-1) + \cdots + p_{m-1} \cdot Pr(A \leq 1) + p_m \cdot Pr(A = 0) \\ &= \sum_{k=0}^{m} p_k \cdot Pr(A \leq m-k) \\ p_j &= p_0 \cdot Pr(A = m+j) + p_1 \cdot Pr(A = m+j-1) + \cdots + p_{m+j-1} \cdot Pr(A = 1) \\ &\quad + p_{m+j} \cdot Pr(A = 0) \\ &= \sum_{k=0}^{m+j} p_k \cdot Pr(A = m+j-k) \quad \text{for } 1 \leq j \end{aligned} \tag{17}$$

which is similar to (7) given in Section 3. Thus, with $A(z) = e^{-\lambda(1-z)}$ as before,

$$P(z) = \frac{K \cdot \Pi_{s=1}^{m} z - z_s}{z^m - A(z)} \tag{18}$$

and K, λ and z_s for $s = 1, 2, \ldots, m-1$ can be solved by following the procedure described in Section 3.

Using the Little's law, the average delay, D, becomes,

$$D = \frac{q^i}{\rho} = \frac{1}{\rho} \cdot \left[\sum_{j=0}^{m-1} \frac{1}{1 - z_j} - \frac{m(m-1) - \lambda^2}{2(m - \lambda)} \right]. \tag{19}$$

For $m = 1$, (19) simplifies to,

$$D = \frac{\rho}{2(1-\rho)} = \frac{1-\rho}{2\rho} + \frac{2\rho - 1}{2\rho(1-\rho)}.$$

For $m > 1$, we observed that $|z_s| \ll 1$ for $s = 1, 2, \ldots, m-1$ for the values of p near 1, and (19) may be approximated to,

$$D \simeq \frac{1}{\rho} \cdot \left[m - 1 - \frac{m(m-1) - \lambda^2}{2(m - \lambda)} \right] = \frac{1-\rho}{2\rho} + \frac{2\rho - 1}{2m\rho(1-\rho)}. \tag{20}$$

We see from (20), that D is finite for $\rho < 1$, and that due to the second term, D is strictly decreasing with increasing m which states the benefit of multi-channel switching. Figure 3 shows samples of calculated delays using (20).

5 Output Buffering

A comprehensive analysis of the output buffering scheme in multi-channel switching is found in [3]. In this section, we omit the detailed analysis of the output buffering scheme and summarize only its essential characteristics for the completeness.

In the context of multi-channel switching, output buffering employs a buffer module for each output channel group and a certain number, v, of the arrived packets at the input can be transmitted to a channel group in a time slot, and be stored in the buffer module until they can find an idle time slot in one of the m output channels belonging to the channel group. We term the factor v as the *speed up* factor for the switching network. We assume again that the size of the buffer modules are infinite.

Let A_n^i be the number of packets arriving at the input in time slot n that are destined for channel group i, and let $\tilde{A}_n^i$ be the number of packets arriving at the buffer module amongst the A_n^i packets arrived at the input. With $v = N$, all arriving packets can reach the output without loss, $\tilde{A}_n^i = \tilde{A}_n^i$. We note that q_n^i, the number of packets in the buffer module for channel group i in time slot n, has the dynamics expressed by,

$$q_{n+1}^i = max(q_n^i + A_n^i - m\ ,\ 0). \tag{21}$$

which is the same as (16) for the shared buffering.

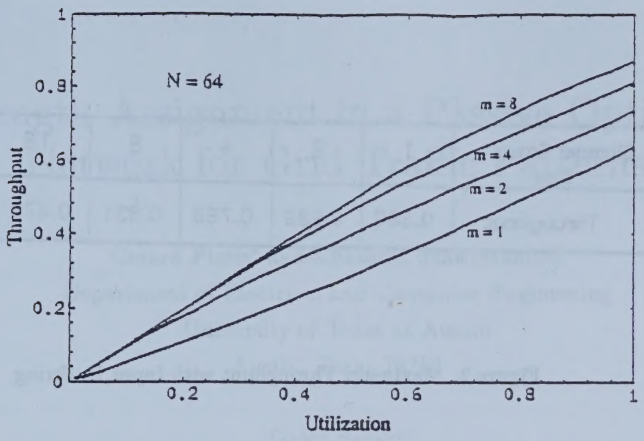


Figure 1: Throughput for Bufferless Network.

The identical argument used in Section 4 illustrates that the maximum throughput for the output buffering equals 1 for all m. The delay, D, is determined by following the same procedure in Section 4 and the benefit of multi-channel switching is asserted by observing the decreasing values of D with respect to the increasing values of m.

If $v < N$,

$$\begin{aligned} \tilde{A}_n^i &= A_n^i \quad \text{if } A_n^i \leq v \\ &= v \quad \text{if } A_n^i > v. \end{aligned}$$

Then, using $\tilde{A}_n^i$ (instead of A_n^i) in (21), the same analysis may be repeated to determine D.

References

[1] Hluchyj, M. and Karol, M., "Queueing in High-Performance Packet Switching," *IEEE J. Selected Areas in Communications*, Vol. 6, No. 9, Dec. 1988, pp. 1587-1597.

[2] Karol, Mark J. and I, Chih-Lin, "Performance Analysis of a Growable Architecture for Broad-Band Packet (ATM) Switching," *IEEE Transactions on Communications*, Vol. 40. No. 2. Feb 1992, pp. 431-439.

[3] Lin, A., and Silvester, J.A., "On the Performance of an ATM Switch with Multichannel Transmission Group," *IEEE Trans. on Communications*, Vol. 41, No. 5, May. 1993, pp. 760-770.

Channel Group Size	1	2	4	8	16
Throughput	0.586	0.686	0.768	0.831	0.878

Figure 2: Maximum Throughput with Input Buffering

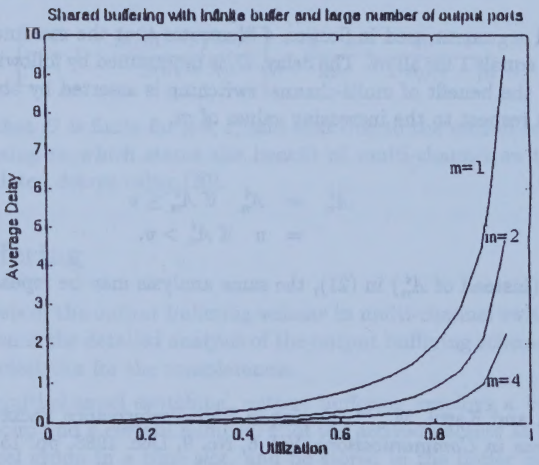


Figure 3: Delay with Shared Buffering

Wavelength Assignment in a Passive Optical Star Network for Grid Traffic Patterns

Gerard Pieris* and Chakarat Skawratanond
Department of Electrical and Computer Engineering
University of Texas at Austin
Austin, Texas 78712

Galen Sasaki*
Department of Electrical Engineering
University of Hawaii
2540 Dole Street Honolulu, Hawaii 96822

Abstract: In this paper, we study the problem of assigning wavelengths to transmitters and receivers in a *wavelength division multiplexed* network to maximize throughput. The network is a passive optical star, where each node has one transmitter and r receivers. It is assumed that the wavelengths are assigned at some initial time, and the assignment is kept thereafter. Communication is constrained so that no *store and forwarding* is possible. In other words, if a node must transfer data to another, it does so directly, without using intermediate nodes. It is also assumed that the traffic pattern is a grid, i.e., the nodes of the network correspond to nodes in the grid and neighboring nodes in the grid communicate with each other at a uniform rate.

We present two node coloring problems for a grid that correspond to the problem of assigning wavelengths to transmitters and receivers to maximize throughput. We provide bounds for the optimal solutions of the problems.

1 Introduction

In this paper, we study the problem of assigning wavelengths to transmitters and receivers in a *wavelength division multiplexed* (WDM) network to maximize throughput. The network is a passive optical star, where each node has one transmitter and r receivers. Each transmitter and receiver has a data rate capacity of c. The communication is constrained so that no *store and forwarding* is possible. In other words, if a node must transfer data to another, it does so directly, without using intermediate nodes. This avoids intermediate electronic processing and the "electronic bottleneck" on throughput.

*This research was supported by the National Science Foundation under grant NCR-9396130 and research funds from TRW.

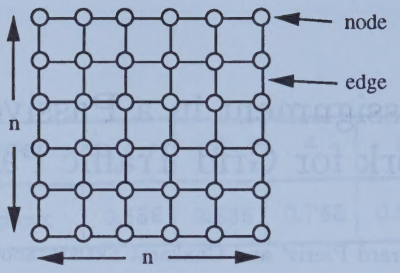


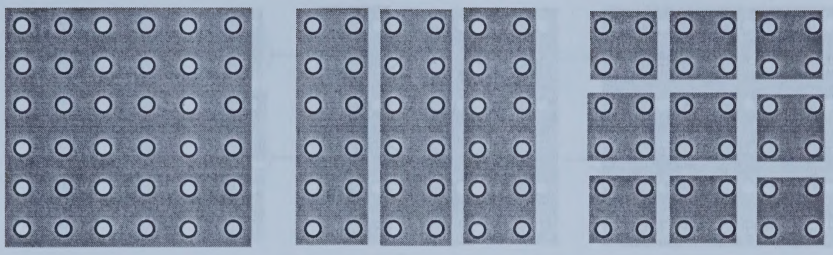
Figure 1: An $n \times n$ grid when $n = 6$.

We assume that the number of nodes is n^2, where n is a positive integer. The traffic pattern corresponds to an $n \times n$ grid, as shown in Figure 1. In the figure, the grid is an $n \times n$ array of dots, where each dot represents a node. A line segment between two dots is called an *edge*. Two nodes with an edge between them are called *neighbors* of each other. For the traffic pattern, it is assumed that neighbors transmit at rate λ to each other, where λ is some nonnegative real valued parameter. The parameter λ is proportional to throughput. In particular, the total throughput is $\lambda \cdot 4[n(n-1)]$, where $2n(n-1)$ is the total number of edges in the grid. This traffic pattern will be called the *grid traffic pattern*. We consider the pattern because it occurs in applications such as in parallel and distributed simulation. Note that wavelength assignment problems have been considered before in [1] [3], but for *uniform* traffic patterns.

We assume that the transmitters and receivers are assigned wavelengths once (at some initial time) and the assignment is fixed thereafter. We also assume that the number of available wavelengths is essentially infinite. The assumptions reflect the fact that transmitters and receivers that can tune to many wavelengths have slow tuning times [2]. Note that wavelengths must be assigned transmitters and receivers so that each node has its receivers tuned to the wavelengths of the transmitters of its neighbors. Such a *wavelength assignment* is referred to as *proper*. A proper wavelength assignment is referred to as *feedback proper* if each node has a receiver assigned the wavelength of its transmitter. The feedback proper property will be required for transmission protocols, such as ALOHA, that use the feedback of its own transmissions.

An assignment of wavelengths to the transmitters of nodes can be represented by assigning colors to the nodes in the grid, where a color corresponds to a wavelength. A node coloring of the grid will be called *proper* if the neighbors of each node have at most r colors. A proper node coloring corresponds to a proper wavelength assignment because each node will have enough receivers to tune to the wavelengths of the transmitters of its neighbors. A node coloring of the grid will be called *feedback proper* if each node and its neighbors have at most r colors. A feedback proper node coloring corresponds to a feedback proper wavelength assignment because each node has enough receivers to tune to the wavelength of its own transmitter as well as the wavelengths of the transmitters of its neighbors.

Given a node coloring, we define the *load* of a color as the number of nodes with the color. The *maximum load* is the maximum of the loads over all colors. Note that if the node coloring corresponds to a wavelength assignment for the transmitters then the maximum load is inversely proportional to the maximum possible value for λ. To be more specific, note that a node u must transmit at a rate $\lambda \cdot d(u)$, where $d(u)$ is the



(a) A node coloring for r = 1. (b) A node coloring for r = 2. (c) A node coloring for r = 3.

Figure 2: Feedback proper node colorings for a 6×6 grid are shown for $r = 1, 2$ and 3. Nodes in a shaded region share the same color.

number of neighbors of u. Hence, the rate at which data is transmitted on a particular wavelength corresponding to color w is $\sum_{u \in h(w)} \lambda \cdot d(u)$, where $h(w)$ denotes the set of nodes colored w. Then $\lambda \leq \frac{c}{\sum_{u \in h(w)} d(u)}$. Therefore, the maximum possible value for λ is $\min_w \frac{c}{\sum_{u \in h(w)} d(u)} = \frac{c}{\max_w \sum_{u \in h(w)} d(u)}$, which we denote by $\lambda_{\max}$. Since a node has between two and four neighbors, $\frac{c}{4 \max_w |h(w)|} \leq \lambda_{\max} \leq \frac{c}{2 \max_w |h(w)|}$. Then $\lambda_{\max} = \delta \frac{c}{\max_w |h(w)|}$, where $1/4 \leq \delta \leq 1/2$ and $\max_w |h(w)|$ is the maximum load.

We consider two node coloring problems. The first is to find a proper node coloring of the $n \times n$ grid that minimizes the maximum load. Let $L(n,r)$ denote the minimum. The second problem is to find a feedback proper node coloring of the $n \times n$ grid that minimizes the maximum load. Let $L_f(n,r)$ denote the minimum. The following are upper bounds for $L(n,r)$ and $L_f(n,r)$.

Theorem 1

$$(a) \qquad L_f(n,r) \leq \begin{cases} n^2, & \text{if } r = 1 \\ 2n, & \text{if } r = 2 \\ 4, & \text{if } r \geq 3 \end{cases}$$

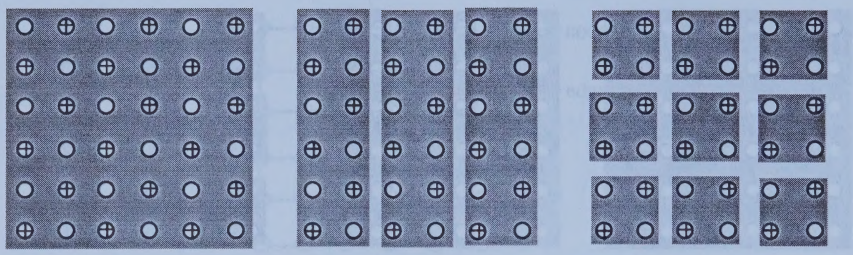
$$(b) \qquad L(n,r) \leq \begin{cases} \lceil n^2/2 \rceil, & \text{if } r = 1 \\ n, & \text{if } r = 2 \\ 2, & \text{if } r \geq 3 \end{cases}$$

The node colorings that lead to the upper bounds in Theorem 1 for $L_f(n,r)$ and $L(n,r)$ are shown in Figures 2 and 3, respectively. Other wavelength assignments can be found in [4].

In Sections 2 and 3, we will provide lower bounds for $L_f(n,r)$ and $L(n,r)$, respectively. In Section 4, we give final remarks.

2 Lower Bounds For $L_f(n,r)$

In this section we will prove the following theorem.



(a) A node coloring for r = 1. (b) A node coloring for r = 2. (c) A node coloring for r = 3.

Figure 3: Proper node colorings for a 6×6 grid are shown for $r = 1, 2$ and 3. Nodes in a shaded region that have the '+' mark share the same color. Nodes in a shaded region and are unmarked share the same color.

Theorem 2

$$L_f(n,r) \geq \begin{cases} n^2, & \text{if } r = 1 \\ n, & \text{if } r = 2 \\ 1, & \text{if } r \geq 3 \end{cases}$$

Before proving the theorem, we will provide preliminary definitions and lemmas. These will be for the case $r = 2$, which is the nontrivial case.

We assume that the $n \times n$ grid is drawn in a square region, as shown in Figure 1. Note that the edges divide the grid into $(n-1) \times (n-1)$ array of subsquares, where each subsquare has nodes at its corners and edges for its sides. The subsquares will be referred to as *local squares* or *L-squares*. A node colored grid is called *locally 2-colored* if, for each L-square, the corner nodes have at most two colors.

Lemma 1 *If $r = 2$ then a feedback proper, node colored grid is locally 2-colored.*

Proof. Consider a feedback proper, node colored grid for $r = 2$. Suppose there is an L-square with at least three different colors at its corner nodes. Let three of the colors be $\{A, B, C\}$. It is easy to check that one of the corner nodes and its neighbors are colored $\{A, B, C\}$, violating the feedback proper property for $r = 2$. Hence, all L-squares have at most two colors at its corner nodes. ■

For a node colored grid, it is assumed, without loss of generality, that the colors are ordered, i.e., if A and B are colors then either $A < B$ or $A > B$. An edge with different colored end nodes is called a *border edge*. The midpoint of a border edge is called a *border point* or *b-point*. If the b-point is located on the side of the grid, it is referred to as an *exterior b-point*. Otherwise, it is referred to as an *interior b-point*.

Given a locally 2-colored grid, we can define closed line segments between b-points in the following way. These line segments will be called *border segments* or *b-segments*. Consider each L-square with b-points, i.e., L-squares that have more than one color at its corner nodes. There are three different types of such L-squares. The first type has one corner node with some color A and other corner nodes with another color B as shown in Figure 4(a). There are two b-points, and a b-segment is the line between them. The

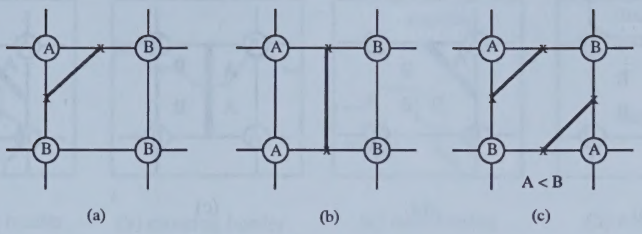


Figure 4: Shown are the b-segments for three types of L-squares. The b-segments are the boldface lines, and the x's are the b-points.

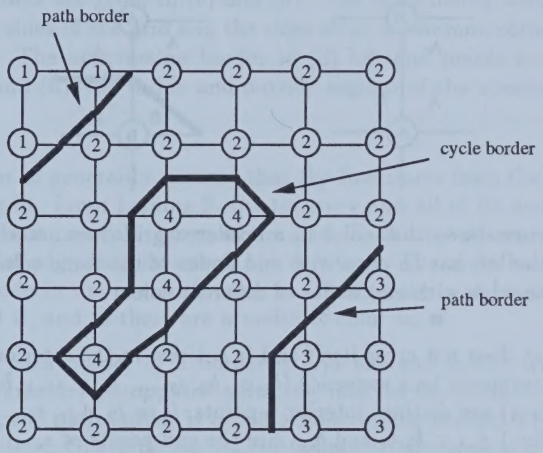


Figure 5: An example of b-segments (in boldface) forming two path borders and a cycle border. The numbers in the nodes represent colors.

second type has two corner nodes with some color A, two corner nodes with another color B, and nodes of the same color are on the same side of the L-square as shown in Figure 4(b). There are two b-points, and a b-segment is the line between them. The third type has two corner nodes with some color A, two corner nodes with another color B, and nodes of the same color are at opposite corners of the L-square as shown in Figure 4(c). Then there are four b-points. Without loss of generality assume $A < B$. There are two b-segments. Each b-segment connects b-points that are in edges sharing a common end node of color A.

The b-segments form a disjoint collection of contiguous sets of b-segments. We call these contiguous sets of b-segments *borders*. Note that each interior b-point has exactly two b-segments incident to it and each exterior b-point has exactly one b-segment incident to it. Thus, there are two types of borders, as shown in Figure 5, which we call *path borders* and *cycle borders*. A path border is a continuous path that does not cross itself and is in the interior of the grid except for its end points. Its end points are at the sides of the grid. More specifically, it corresponds to a sequence $(b_1, s_1, b_2, s_2, ..., b_{k-1}, s_{k-1}, b_k)$, for some $k > 0$, where $(b_1, b_2, ..., b_k)$ are distinct b-points, b_1 and b_k are exterior b-points, $(b_2, b_3, ..., b_{k-1})$ are interior b-points, $(s_1, s_2, ..., s_{k-1})$ are distinct b-segments, and for $1 \leq i < k$, b_i and b_{i+1} are the end points of s_i. A cycle border is a continuous

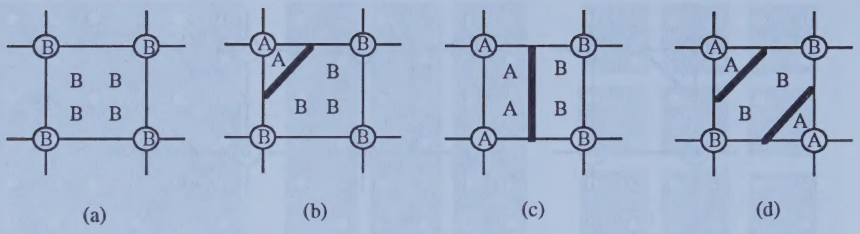


Figure 6: Four types of L-squares are shown with their regions shaded.

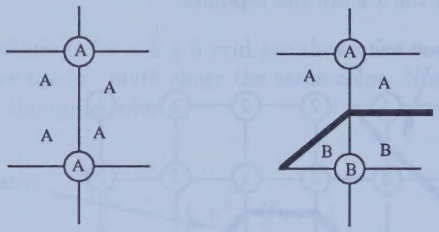


Figure 7: This figure shows that color in a bordered grid does not change across edges. The example on the left has an edge with end nodes of the same color, and the example on the right has an edge with end nodes of different color.

loop of a path that does not cross itself and is entirely in the interior of the grid. More specifically, it corresponds to a sequence $(b_1, s_1, b_2, s_2, ..., b_{k-1}, s_{k-1}, b_k)$, for some $k > 0$, where $(b_1, b_2, ..., b_{k-1})$ are distinct interior b-points, $b_1 = b_k$, $(s_1, s_2, ..., s_{k-1})$ are distinct b-segments, and for $1 \leq i < k$, b_i and b_{i+1} are the end points of s_i.

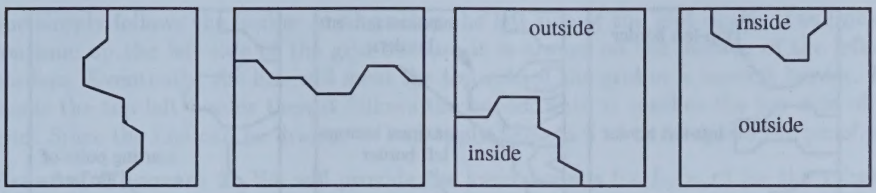
A locally 2-colored grid with its b-segments can be *shaded* as follows. Note that if an L-square has b-segments then the L-square region is partitioned by its b-segments into subregions. The subregions are colored (or shaded) with the same color as the corner nodes that it contains, as shown in Figure 6. If an L-square has no b-segments then its region is colored the same as its corner nodes. A locally 2-colored grid with its b-segments and with each of its L-squares shaded is called a *bordered grid.*

Note that a bordered grid is partitioned by its borders into disjoint regions. We will refer to these regions as *territories.*

Lemma 2 *Consider a bordered grid. Any territory and its nodes are of one color.*

Proof. Note that within an L-square, the color changes only at the borders (see Figure 6). Also note that for L-squares sharing a common edge, color changes only across the borders and not across the edges (see Figure 7). Since the color changes only across the borders, the region within a territory is of one color. In addition, a node has the same color as its surrounding region, so the nodes in a territory have the same color as the territory. ■

Lemma 3 *Consider a bordered grid. Suppose you can draw a line (not necessarily straight) that connects two opposite sides of the grid and is within the interior of a territory except for its end points. Then there are n nodes of the same color.*



(a) crossing border (b) crossing border (c) noncrossing border (d) noncrossing border

Figure 8: Two examples of crossing borders are given in (*a*) and (*b*), and two examples of noncrossing borders are given in (*c*) and (*d*). The noncrossing border in (*c*) has end points on different sides of the grid and the sides share a common corner of the grid (the lower left corner). The noncrossing border in (*d*) has end points on the same side of the grid. For (*c*) and (*d*), the inside and outside regions of the noncrossing borders are shown.

Proof. Without loss of generality, assume that the line starts from the bottom and ends at the top of the grid. From Lemma 2, the territory and all of its nodes are of a single color w. The line must cross all n rows of the grid. Each time it crosses a row it crosses a node or horizontal edge that intersects the territory. The node has color w, and the edge has an end node in the row with color w. Thus, there is at least one node in each row that is colored w, and so there are n nodes of color w. ■

Recall that a path border of a bordered grid has end points on the sides of the grid. If the end points are on opposite sides, we refer to the path border as a *crossing border*. Otherwise, the path border is called a *noncrossing border*. There are two types of noncrossing borders. One type has its end points at the same side of the grid. The other type has its end points at different sides of the grid, but the sides share a common corner node of the grid. See Figure 8 for examples of crossing and noncrossing borders.

Note that a noncrossing border partitions the grid into two regions, as shown in Figures 8(c) and (d). One region contains less than two corner nodes of the grid, and the other region contains more than two corner nodes. We will refer to the region that contains less than two corner nodes as the *inside* of the border, and refer to the other region as the *outside* of the border.

Lemma 4 *A bordered grid with a crossing border has n nodes of the same color.*

Proof. Without loss of generality, assume that the crossing border goes from the bottom to the top of the grid, such as in Figure 8(a). Then we can draw a line (not necessarily straight), just to the right of the border, that starts at the bottom and follows the border to the top. The line can be drawn without crossing any borders because the borders are disjoint. The line can be drawn in the interior of the grid because the border is in the interior except for its end points. Lemma 3 and the existence of the line imply that there are at least n nodes of the same color. ■

Lemma 5 *A bordered grid with no crossing borders has n nodes of the same color.*

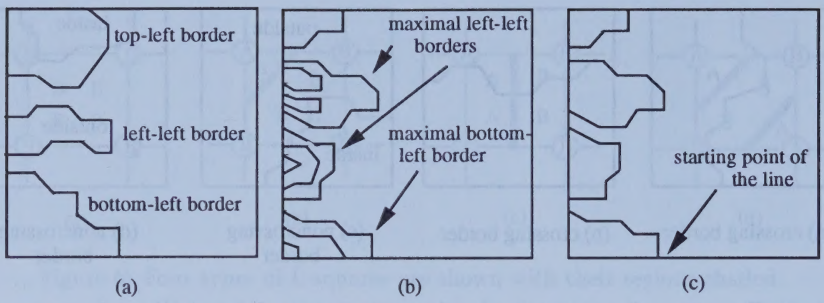


Figure 9: Examples of top-left, left-left, and bottom-left borders are given in (a). Examples of maximal borders are given in (b). An example set of bottom-left and left-left borders is given in (c) under the assumption that the only left-left borders are the maximal ones outside the maximal bottom-left border, and the only bottom-left border is the maximal one. The starting point of the line is also shown in (c).

Proof. Consider a bordered grid with no crossing border, i.e., all of its path borders are noncrossing. In this proof we will show that we can draw a line (not necessarily straight) that connects the bottom and the top sides of the grid and is entirely within the interior of a territory except for its end points. Then we can apply Lemma 3 to complete the proof.

The line that we consider will follow the left side of the grid as well as noncrossing borders connected to the left side. There are three types of noncrossing borders connected to the left side which we call the *top-left*, *left-left*, and *bottom-left* borders. A top-left border is connected to the top and left sides, a left-left border is connected to the left side only, and a bottom-left border is connected to the bottom and left sides. See Figure 9(a) for examples.

Note that a border is either on the inside or the outside of another border because borders are disjoint, We can call a left-left border *maximal* if it is on the outside of the other left-left borders. Thus, any a left-left border that is not maximal is on the inside of some maximal left-left border. Similarly, we can call a bottom-left border *maximal* if it is on the outside of all other bottom-left borders. There is only one maximal bottom-left border because the insides of all bottom-left borders contain the bottom-left corner node of the grid. Figure 9(b) shows examples of maximal borders.

The line we consider will start at the bottom of the grid and will not cross any borders. If there is a maximal bottom-left border then the line will start just to the right of it. Otherwise, the line starts just to the right of the left side of the grid. Note that the line starts on the outside of the maximal bottom-left and maximal left-left borders. In addition, it never crosses a border. Thus, we can assume without loss of generality that the only left-left borders are the maximal ones that are outside of the maximal bottom-left border. We can also assume without loss of generality that the only bottom-left border is the maximal one. The result of the assumptions is shown in Figure 9(c).

Note that the left-left and bottom-left borders are outside of each other. Therefore, the line can be drawn as follows. The line follows the bottom-left border (or left side of the grid if there is no bottom-left border) until it reaches the left side of the grid. The line follows the left side upwards. If the line encounters a left-left border then the

line simply follows the border until reaches the left side of the grid again. The line can continue up the left side of the grid because it is always on the outside of the left-left borders. Eventually, the line will meet the top side of the grid or a top-left border. If it meets the top-left border then it follows the border until it reaches the top side of the grid. Since the line can be drawn, we can apply Lemma 3 to to complete the proof. ■

Proof of Theorem 2. We will provide the lower bounds for $L_f(n,r)$ for the values of $r = 1,2$, and 3. For the case $r = 1$, consider a feedback proper, node coloring of the $n \times n$ grid. Since the coloring is feedback proper, each node and its neighbors must have the same color. Since the grid is a connected graph, all nodes must have the same color. Thus, $L_f(n,1) \geq n^2$.

For the case $r = 2$, consider a feedback proper, node coloring of the $n \times n$ grid. Then Lemma 1 implies that there is a corresponding bordered grid with the same colors assigned to the nodes. Lemmas 4 and 5 imply there are at least n nodes with the same color. Thus, $L_f(n,2) \geq n$.

For the case $r = 3$, the lower bound $L_f(n,3) \geq 1$ is trivially true because each node has a color. ■

3 Lower Bounds For $L(n,r)$.

Lower bounds for $L(n,r)$ are given in the following theorem.

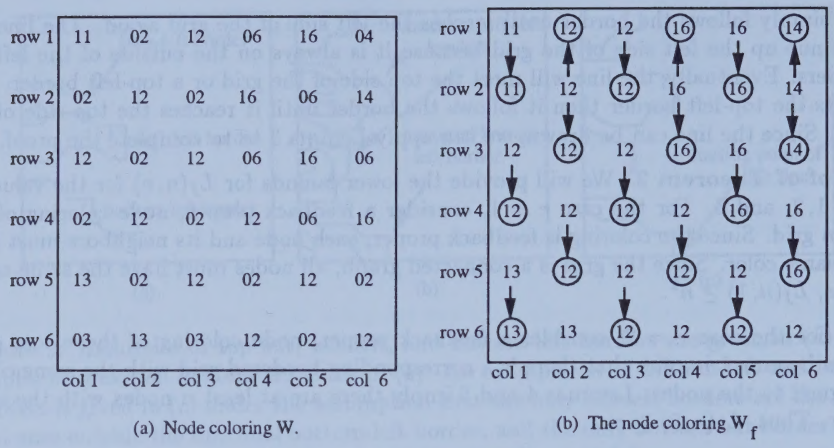
Theorem 3

$$L(n,r) \geq \begin{cases} n^2/2 - n/4, & \text{if } r = 1 \\ n/3, & \text{if } r = 2 \\ 1, & \text{if } r \geq 3 \end{cases}$$

The theorem is implied by Theorem 2 and the following lemma.

Lemma 6 *Consider a proper node coloring W that has maximum load k. Then there is a feedback proper node coloring W_f that has maximum load at most* $\min\{3k, 2k + n/2\}$.

The lemma can be verified by using the following procedure that transforms a proper node coloring W into a feedback proper node coloring W_f. The procedure is shown in Figure 10. The node coloring W is shown in Figure 10(a). The rows of the grid are numbered 1 to n from top to bottom, and the columns of the grid are numbered 1 to n from left to right. Nodes in odd rows and even columns and nodes in even rows and odd columns are *tagged.* The tagged nodes are circled in Figure 10(b). A tagged node in row 1 replaces its color with the color of the node directly below it. A tagged node that is not in row 1 replaces its color with the color of the node directly above it. The arrows in Figure 10(b) show the direction in which colors are copied. Figure 10(b) shows the new coloring W_f. It is straightforward to check that W_f is feedback proper.



(a) Node coloring W.

(b) The node coloring W_f

Figure 10: This figure illustrates the procedure that transforms a proper node coloring W to a feedback proper node coloring W_f.

4 Final Remarks

It would be of interest to tighten the bounds for $L(n,r)$ and $L_f(n,r)$. We suspect that the upper bound for $L(n,r)$ could be improved. It would also be of interest to extend our results to 3-dimensional grid traffic patterns. In [4], wavelength assignments were given that lead to lower bounds on throughput for 3-dimensional grid traffic patterns. However, it remains to find good upper bounds.

References

[1] Y. Birk, and F. A. Tobagi, and M. E. Marhic, "Bus-oriented Interconnection Topologies for Single-Hop Communication Among Multi-transceiver Stations," *IEEE INFOCOM '88 Proceedings*, New Orleans, Louisiana, March 1988, pp. 558-567.

[2] Charles A. Bracket, "Dense Wavelength Division Multiplexing Networks: Principles and Applications," *IEEE J. Sel. Areas Comm.*, vol. 8, no. 6, Aug. 1990, pp. 948-964.

[3] M. Busche and B. Hajek, "On Optical Interconnection of Stations Having Multiple Transmitters and Receivers," *Proc. 1990 Int. Symp. Info. Theory and Its Appl.*, Honolulu, Hawaii, Nov. 1990, pp. 967-970.

[4] C. Skawratananond, "Design methodology for bus-oriented networks with single-hop constraint," The University of Texas at Austin, Master's Thesis, August, 1992.

A NOVEL ADAPTIVE IIR ACOUSTIC ECHO CANCELLER

M. Tahernezhadi, L. Liu, and G. Miller
Department of Electrical Engineering
Northern Illinois University
DeKalb, Il 60115

Abstract- In this paper, a pole-zero lattice adaptive *acoustic echo canceller* (AEC) for teleconferencing application is proposed. The proposed algorithm utilizes the lattice and inverse lattice (known as LATIN) configuration proposed by Chao and Tsujii in order to separate the near-end speech without any distortion from the acoustic echo signal in the *double-talk* (DT) mode. Considering a structure with M poles and M zeros, the complexity of the proposed algorithm is approximately twice the complexity of an M-tap FIR gradient lattice algorithm.

I. INTRODUCTION

In recent years, there has been a growing interest in acoustic echo cancellation for speaker phone and teleconferencing applications. Acoustic echo is the dominant culprit that degrades the speech quality in such applications. The reverberation of a teleconferencing room results in a very long acoustic echo tail, the acoustic echo could last up to hundreds of milliseconds. As a result, the length of the impulse response of acoustic echo path often exceeds thousands sampling period. For example, a filter length of 3200 is required to cancel 400ms acoustic echo when the sampling rate is 8kHz. The modeling of this kind of echo path not only requires enormous FIR adaptive filter, but also increases the number of memory locations and number of operations. Moreover, the acoustic echo path is generally non-stationary. A successful echo canceller should have the ability to trace the change of acoustic echo quickly enough to cancel the echo, therefore, a fast convergence is required. As is well known, the LMS algorithm converges poorly when input signal has large eigenvalue spread and when the length of filter is large. This precludes the application of LMS-type algorithm as an *acoustic echo canceller* (AEC) when a large number of taps is required.

These problems have motivated other approaches with the double purpose of reducing the computational complexity and of improving the convergence speed of the algorithm. Many studies are carried out on transversal filters due to their simplicity and stability. The RLS algorithm converges well but the computation cost is high [1]. The most mature and successful scheme is subband filtering [2]. However, even with subband techniques, the massive number of taps will be costly. Recently IIR pole-zero structures [3] are proposed for efficient echo cancellation in systems whose impulse response continues for several thousands sample periods. It has been reported that the IIR structures of AEC can be much more efficient than FIR ones [4].

The acoustic echo path is inherently an infinite impulse response system. The potential of computational saving of IIR filter is obvious, because the combined number of taps required in the denominator and numerator is greatly less than that of a FIR filter. The stability requirements always come with the implementation of an IIR adaptive filter. SHARF (Simple Hyperstable Adaptive Recursive Filter) algorithm [5] and pole-zero structure [3] are proposed to increase the stability of IIR system. The pole-zero structure is shown in Fig. 1. As seen, the adaptation uses a different error signal $e'(n)$ as opposed to the final error signal e(n). Referring to Fig. 1, one can obtain error e(n) from $e'(n)$ as $E(z) = E'(z)/B(z)$. From a practical point of view, one can send out $e'(n)$ as the final error during the single talk periods, thus obviating an IIR filtering stage. During double talk periods, i.e., when both far-end and near-end signals are present, the acoustic echo will be imbedded in the interference speech due to near-end signal. As such, to avoid distortion of near-end speech signal, the IIR filter 1/B(z) has to be supplied before $e'(n)$ is sent out.

To implement an AEC with IIR filter, convergence rate and stability have to be considered. Lattice structures are of concern due to their fast convergence and inherent stability. The IIR echo canceller proposed by Chao and Tsujii [4]-[5] uses a two-dimensional lattice structure. The stability requirement for this IIR lattice is that the eigenvalues of all the partial correlation matrices are inside the unit circle. This condition can be satisfied for a multichannel maximum entropy (ME) lattice filter, which has an enormously computational cost. A simpler lattice filter [4], which is updated by the *Levinson-Wiggins-Wittle-Robinson* (LWR) algorithm, needs additional gains to ensure the stability. Even with the LWR algorithm, the computational complexity of the two-dimensional lattice is high when the number of lattice stages is increased.

In this paper, a new one dimensional IIR lattice algorithm is proposed in order to avoid the two-dimensional lattice. The proposed algorithm passes the two signals x(n) and d(n) through a one-dimensional lattice at each iteration. Thus, the computation complexity is only twice the ordinary lattice filter.

II. The proposed Algorithm

Consider the echo canceller structure in Fig. 1 with number of AR and MA parameters assumed to be equal to M. The MA parameters, a_i's, and the AR parameters, b_i's, need to be updated as to minimize the mean square error of $e'(n)$ given by

$$e'(n) = d(n) - \sum_{i=1}^{M} (b_i d(n-i) - a_i x(n-i)) \tag{1}$$

Clearly, $e'(n)$ can be regarded as forward prediction error of d(n) for an (M,M) ARMA model. Likewise, one can define the forward prediction error of x(n) for an (M,M) ARMA process as

$$e''(n) = x(n) - \sum_{i=1}^{M} (q_i x(n-i) - p_i d(n-i)) \tag{2}$$

where the MA parameters p_i's and AR parameters q_i's are updated as to minimize the mean square value of $e''(n)$. At this point let us define a new sequence s(n) constructed from x(n) and d(n) as

$$s(2n) = x(n) \qquad s(2n+1) = d(n) \tag{3}$$

That is, the even and odd indices of s(n) refer to x(n) and d(n), respectively. The $e'(n)$ and $e''(n)$ can now be expressed in terms of s(n) as

$$e'(n) = s(2n+1) - \sum_{i=1}^{2M} u_i s(2n+1-i) \tag{4a}$$

$$e''(n) = s(2n) - \sum_{i=1}^{2M} v_i s(2n-i) \tag{4b}$$

where

$$u_{2i} = b_i \qquad u_{2i-1} = a_i \tag{5a}$$

$$v_{2i} = q_i \qquad v_{2i-1} = p_i \tag{5b}$$

Thus, $e'(n)$ and $e''(n)$ are the linear prediction error of s(n) at indices 2n+1 and 2n, respectively. That is, the problem has been transformed into a one dimensional linear prediction problem which can be solved using a one dimensional gradient lattice filter [6] shown in Fig. 2.

The lattice structure shown in Fig. 2 is characterized by the output-input relations given as

$$f_m(n) = f_{m-1}(n) - k_m^f(n) b_{m-1}(n-1) \tag{6a}$$

$$b_m(n) = b_{m-1}(n-1) - k_m^b(n) f_{m-1}(n) \tag{6b}$$

As opposed to conventional lattice filter for one dimensional case, the backward and forward reflection coefficients, $k_m^f(n)$ and $k_m^b(n)$, are allowed to be different because the stochastic parameters of s(n) vary from odd indices to even indices. The $k_m^f(n)$ and $k_m^b(n)$ are optimized according to least-square criterion. For example, the forward reflection coefficient at odd indices, $k_m^f(2n+1)$ are chosen to minimize the cost function $E_m^f(2n+1)$ given by

$$E_m^f(2n+1) = \sum_{l=0}^{n} w^{n-l} |f_m(2l+1)|^2$$

$$= \sum_{l=0}^{n} w^{n-l} |f_{m-1}(2l+1) - k_m^f(2n+1) b_{m-1}(2l)|^2 \tag{7}$$

It can be readily shown that the minimization of (7) with respect to $k_m^f(2n+1)$ leads to the following recursive time updation for reflection coefficients at odd time indices

$$k_m^f(2n+1) = k_m^f(2n-1) + \frac{b_{m-1}(2n) f_m(2n+1)}{\eta_m(2n+1)} \tag{8}$$

where

$$\eta_m(2n+1) = w\eta_m(2n-1) + |b_{m-1}(2n)|^2 \tag{9}$$

Likewise, the above least-square based recursive time updation can be extended to determine the time updation of even index forward reflection coefficients as well as backward reflection coefficients at odd and even indices. Hence, one arrives at the complete forward lattice algorithm shown in Table 1. It is seen that the error $e'(n)$ is obtained as the last stage forward prediction error at the odd index 2n+1, i.e., $f_{2M}(2n+1)$. The error $e''(n)$ is not required to be sent out in the AEC applications.

It should be noted that the proposed ARMA algorithm is equivalent to two dimensional approach presented in [7]. At each iteration, the two dimensional lattice determined by the 2x2 reflection coefficient matrix, while the proposed one dimensional lattice uses the forward and backward prediction coefficients at even and odd indices.

As mentioned before, whereas in the absence of near-end signal, the desired output is $e'(n)$, during the double talk period the algorithm output should be e(n), which is $e'(n)$ filtered by an all pole filter 1/B(z). Since b_i's are equal to the already computed coefficients of U(z) at even indices (see (5a)), the filter 1/B(z) can be realized by an inverse lattice as follows.

If the transfer function between $f_o(2n+1) = d(n)$ and $f_{2M}(2n+1) = e'(n)$ is $U_{2M}(z) = 1 - \sum_{i=1}^{2M} u_i z^{-i}$, then the transfer function between $f_{2M}(2n+1)$ and $f_o(2n+1)$ is $1/U_{2M}(z)$. That is,

$$f_o(2n+1) = f_{2M}(2n+1) + \sum_{i=1}^{M} b_i f_o(2n-2i+1) + \sum_{i=1}^{M} a_i f_o(2n-2(i-1)) \tag{10}$$

Setting all $f_o(n)$ at even indices to zero as to do only 1/B(z) filtering, (10) becomes

$$f_o(2n+1) = f_{2M}(2n+1) + \sum_{i=1}^{M} b_i f_o(2n-2i+1) \tag{11}$$

As a result, e(n) is equal to $f_o(2n+1)$. Exploiting (6), it becomes evident that the direction of calculation at odd indices is from stage 2M to stage 0; and at even indices, the calculation is according to the forward lattice relation, i.e., 0 to 2M. The complete structure of the inverse-lattice structure for the calculation of e(n) is shown in Table 2. $f'(n)$ and $b'(n)$ are used to represent the forward and backward prediction errors in inverse lattice structure. The complete block diagram for the proposed IIR AEC is shown in Fig. 3.

III. Results and Discussions

The impulse and frequency response of acoustic echo used for simulation are shown in Fig. 4. Assuming no error, the error signal at the output of AEC, $e'(n)$, for white noise input and speech input are illustrated in Fig. 5 and 6. An extremely fast convergence for the proposed algorithm can be observed. The performance of the proposed IIR AEC algorithm (30 poles, 30 zeros), the FIR gradient lattice-ladder algorithm (500 taps), and the two-dimensional IIR lattice AEC (30 poles, 30 zeros) [4] are shown in Fig. 7 for a speech input in terms of *echo return loss enhancement (ERLE)*, defined as the ratio of the original echo power level to the residual echo power level in decibel [8]. An ERLE of 20dB and a good convergence rate are seen for both one and two dimensional IIR lattices. For FIR structure, even with lattice algorithm and a relatively long filter length (500), the performance is not good as the IIR structures. It is also seen that the ERLE of the two-dimensional lattice structure is a little superior to the proposed algorithm. Nevertheless, the saving in computation as well as the storage makes the proposed algorithm more suitable in real time implementation of AEC.

Fig. 8 shows the performance of the proposed IIR AEC during double talk, when there exists the interference of near-end speech. It is important to obtain a distortion-free residual echo for the AEC. Otherwise, the near end speech will be distorted. The proposed inverse lattice algorithm is used to mitigate distortion. Fig. 8a shows the acoustic echo of the far end speech and Fig. 8b shows the near end speech being introduced at iteration 5000. Suppose the double talk condition is detected after 10 iterations and the updation of the reflection coefficients is frozen at that time. Figure 8(c) shows $e'(n)$. It can be seen that the near end speech has been significantly distorted. A distortion-free error signal, e(n), is obtained by use of the inverse lattice algorithm. The result is shown in Fig. 8d. Compared to the original near end speech signal in Fig. 8b, exhibits no noticeable distortion while the acoustic echo has been greatly attenuated.

REFERENCES

[1] J. Makhoul, J., "A class of all-zero lattice digital filters: properties and applications," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, no. 4, pp. 791-802, 1984.

[2] Gilloire, A. and Vetterli, M., "Adaptive filtering in subbands with critical sampling: analysis, experiments, and application to acoustic echo cancellation," *IEEE Trans. Signal Processing*, vol. 40, no. 8, pp. 1862-1875. Aug. 1992

[3] Long, D., Shewed, D. and Falconer, D. D., "Study of a pole-zero adaptive echo canceler," *IEEE Trans. Circuits Syst.*, vol. CAS-34, no. 4, Aug 1982.

[4] Chao, J. and Tsujii, S., "A stable and distortion-free IIR echo and howling canceler," *IEEE Trans. Signal Processing*, vol. 39, no. 8, pp. 1812-1821, Aug. 1991.

[5] Chao, J. and Tsujii, S., "A new configuration for echo canceler adaptable during double talk periods," *IEEE Trans. Comm.*, vol. 37, no. 9, pp. 969-947, Sept. 1989.

[6] J. G. Proakis, *Digital Communications*, New York: McGraw-Hill Inc., 1983.

[7] Lee, D. T. L., Friedlander, B. and Morf, M., "Recursive ladder algorithms For ARMA modeling," *IEEE Trans. Autom. Control*, vol. AC-27, no. 4, pp. 753-764, Aug. 1982.

[8] Sondhi, . M. and Berkley, D. A., "Silencing echoes on the telephone network," in *Proc. IEEE*, vol. 68, no.8, Aug. 1980, pp. 948-963

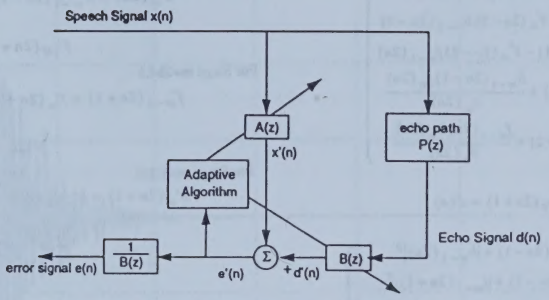
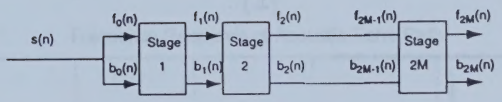


Figure 1. New Pole -Zero Structure Adaptive Echo Canceller



Stage m

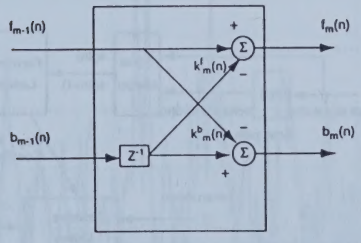


Figure 2. Lattice Structure for Linear Prediction

Table 1. IIR Lattice Algorithm (Forward Lattice)

$$f_o(2n) = b_0(2n) = x(n)$$

For Stage m=1:2M

$$\begin{pmatrix} \eta_m(2n) = w\eta_m(2n-2) + |b_{m-1}(2n-1)|^2 \\ \theta_m(2n) = w\theta_m(2n-2) + |f_{m-1}(2n)|^2 \\ f_m(2n) = f_{m-1}(2n) - k^f_m(2n-2)\, b_{m-1}(2n-1) \\ b_m(2n) = b_{m-1}(2n-1) - k^b_m(2n-2) f_{m-1}(2n) \\ k^f_m(2n) = k^f_m(2n-2) + \dfrac{b_{m-1}(2n-1) f_m(2n)}{\eta_m(2n)} \\ k^b_m(2n) = k^b_m(2n-2) + \dfrac{f_{m-1}(2n)\, b_m(2n)}{\theta_m(2n)} \end{pmatrix}$$

$$f_o(2n+1) = b_0(2n+1) = d(n)$$

For Stage m=1:2M

$$\begin{pmatrix} \eta_m(2n+1) = w\eta_m(2n-1) + |b_{m-1}(2n)|^2 \\ \theta_m(2n+1) = w\theta_m(2n-1) + |f_{m-1}(2n+1)|^2 \\ f_m(2n+1) = f_{m-1}(2n+1) - k^f_m(2n-1)\, b_{m-1}(2n) \\ b_m(2n+1) = b_{m-1}(2n) - k^b_m(2n-1) f_{m-1}(2n+1) \\ k^f_m(2n+1) = k^f_m(2n-1) + \dfrac{b_{m-1}(2n) f_m(2n+1)}{\eta_m(2n+1)} \\ k^b_m(2n+1) = k^b_m(2n-1) + \dfrac{f_{m-1}(2n+1)\, b_m(2n+1)}{\theta_m(2n+1)} \end{pmatrix}$$

$$e'(n) = f_{2M}(2n+1)$$

Table 2. IIR Lattice Algorithm (Inverse Lattice)

$$f'_o(2n) = b'_0(2n) = 0$$

For Stage m=1:2M

$$\begin{pmatrix} f'_m(2n) = f'_{m-1}(2n) - k^f_m(2n-2)\, b'_{m-1}(2n-1) \\ b'_m(2n) = b'_{m-1}(2n-1) - k^b_m(2n-2) f'_{m-1}(2n) \end{pmatrix}$$

$$f'_{2M}(2n+1) = e'(n)$$

For Stage m=2M:1

$$f'_{m-1}(2n+1) = f'_m(2n+1) + k^f_m(2n-1)\, b'_{m-1}(2n)$$

$$b'(0) = f'(0)$$

For Stage m=1:2M

$$b'_m(2n+1) = b'_{m-1}(2n) - k^b_m(2n-1) f'_{m-1}(2n+1)$$

$$e(n) = f'_o(2n+1)$$

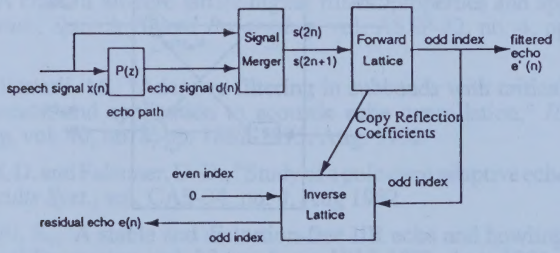


Figure 3. A New Configuration of IIR Lattice AEC

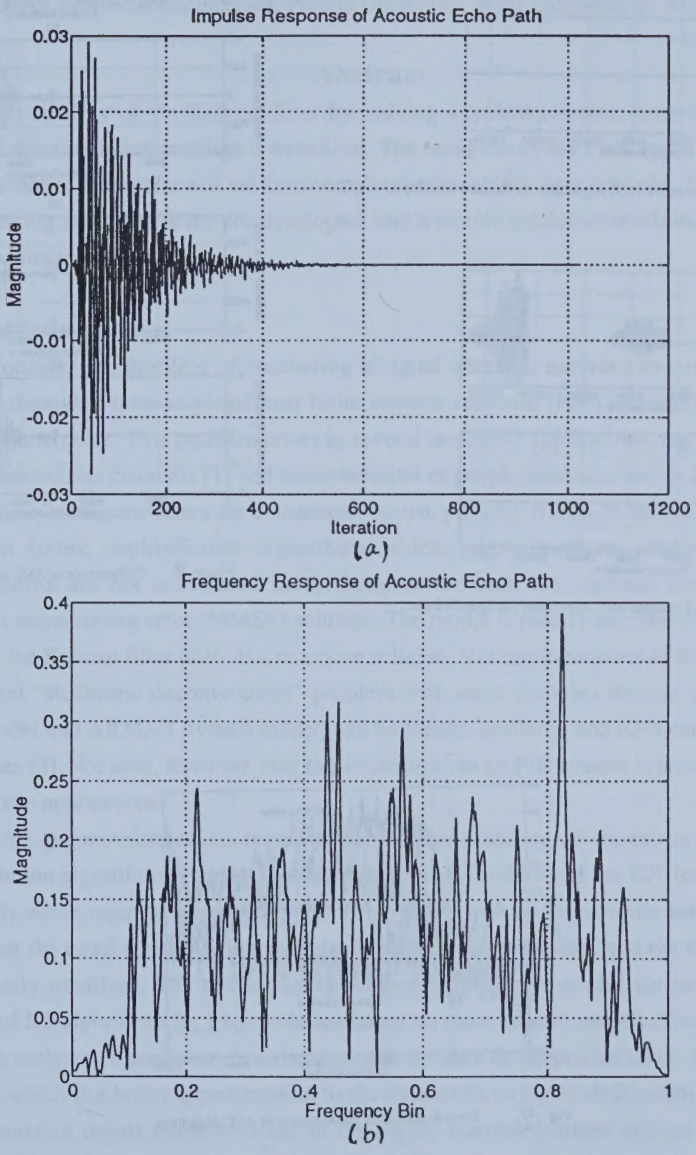


Figure 4. Response of Acoustic Echo Path

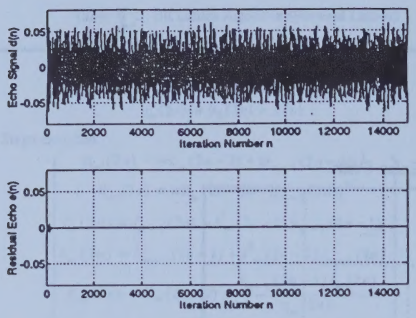


Figure 5. Attenuation of Proposed AEC Algorithm for White Noise Input

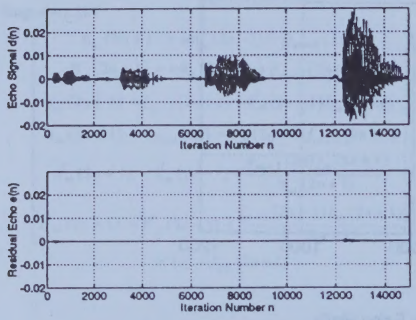


Figure 6. Attenuation of Proposed AEC Algorithm for Speech Input

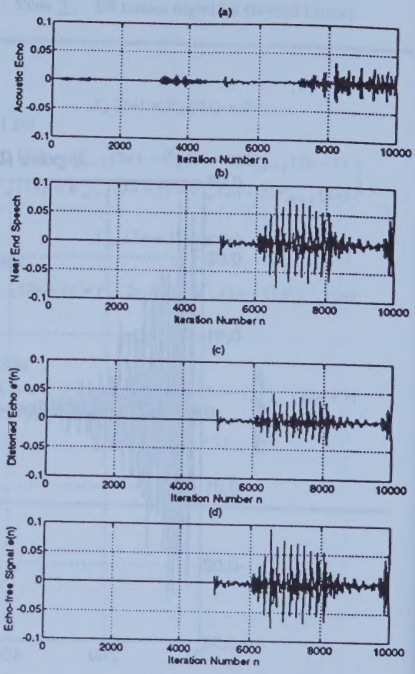


Figure 8 Performance of AEC during Double Talk

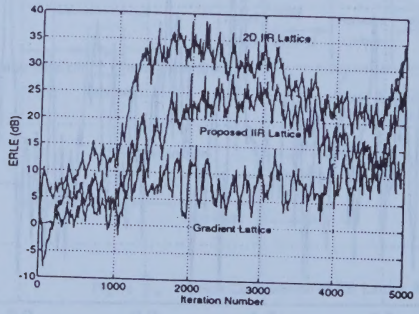


Fig 7. Error Return Loss Enhancement of AEC Algorithm

Stability of Kalman Filtering Algorithms for Stochastic Deconvolution Problem

Saul B. Gelfand[1] **and James V. Krogmeier**
School of Electrical Engineering, Purdue University, West Lafayette, IN 47907-1285

Abstract

The stability of the Kalman filter for solving a typical (discrete-time, time-varying) stochastic deconvolution problem is examined. The usual theory does not apply as the system transition matrix is singular and uniform complete observability does not hold. Techniques for circumventing these problems are developed, and a simple sufficient condition is given for uniform asymptotic stability.

1. Introduction

Consider the problem of recovering a signal which is modeled as a white process observed through a time-varying linear finite impulse response (FIR) system (channel) with additive white noise. This problem arises in several important applications, e.g., equalization of communications channels [1] and deconvolution in geophysical time series [2] . Here we assume the system parameters are considered known, possibly from a recursive identification algorithm (more sophisticated algorithms which jointly perform identification and deconvolution are not considered here). Suppose we seek the optimal fixed lag linear minimum mean square error (MMSE) solution. The model is readily expressed in state-space form and the Kalman filter (KF) is a recursive solution. Various extensions of this approach to the general "stochastic deconvolution" problem with more complex models (e.g., ARMA signal model and ARMAX system model) can be treated similarly, and KF's can be found for these cases [3]. We note, however, that the white input to an FIR system is encountered most frequently in applications.

This paper initiates an investigation of the stability of model-based recursive deconvolution algorithms. Here we examine in detail the stability of the KF for recovering a (modeled) white input to a time-varying FIR system with additive white noise. It will be shown that the usual approach to proving such stability is inadequate, and the theory must be significantly modified. The techniques developed here may be useful for establishing the stability of KF algorithms for deconvolution based on more general models. They may also be helpful in analyzing a nonlinear recursive structure we have developed called the Bayesian filter (BF) [4] which is a better approximation to the optimal fixed lag MMSE solution for the case of nonGaussian inputs (such as occur in the digital communications system with discrete inputs, and the seismic pulse compression with nonGaussian reflectivity series).

[1] Research partially supported by the National Science Foundation under grant NCR-9406073

2. Main Result and Discussion

Let d_k, k = -K,-K+1,..., be an independent sequence of random variables where d_k has mean zero and variance $q_k > 0$. Suppose we observe d_k through an FIR system (channel) with additive noise

$$y_k = \sum_{i=0}^{K} c_i(k) d_{k-i} + v_k, \qquad k = 0,1,...$$

where the impulse response coefficients $c_i(k)$ are known, and the noise v_k, k = 0,1,..., is an independent sequence with mean zero and variance $r_k > 0$ (and $\{d_k\}$ and $\{v_k\}$ are jointly independent). Let $Y(k) = \{y_0,...,y_k\}$ denote the observations up to time k. We seek the fixed lag linear MMSE estimate of d_k, i.e., an estimate of d_{k-K} based on Y(k) (we allow that $c_i(k)$ can be zero so that lags larger than the channel length are allowed).

A state-space parameterization of this model is

$$x(k+1) = Ax(k) + Gw_k \tag{1a}$$

$$y_k = H(k)x(k) + v_k, \qquad k = 0,1,... \tag{1b}$$

where

$$x(k) = [d_k \cdots d_{k-K}]^T, \quad w_k = d_{k+1},$$

$$A = \begin{bmatrix} 0 & & 0 \\ 1 & \ddots & \\ & \ddots & \ddots \\ 0 & 1 & 0 \end{bmatrix}, \qquad G = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \qquad H(k) = [c_0(k) \cdots c_K(k)]$$

Denoting by $\hat{x}(i \mid j)$ and $P(i \mid j)$ the linear MMSE estimate of x(i) given Y(j) and its error covariance matrix, respectively, the KF equations are

$$\hat{x}(k \mid k) = (I - K(k)H(k))A\hat{x}(k-1 \mid k-1) + K(k)y_k \tag{2a}$$

$$P(k \mid k) = P(k \mid k-1) - K(k)H(k)P(k \mid k-1) \tag{2b}$$

$$P(k \mid k-1) = AP(k-1 \mid k-1)A^T + GG^T q_k \tag{2c}$$

where

$$K(k) = P(k \mid k-1)H^T(k)(H(k)P(k \mid k-1)H^T(k) + r_k)^{-1}$$

and $\hat{x}(0 \mid 0) = K(0)y_0$ and $P(0 \mid -1) = \mathrm{diag}(q_0,...,q_{-K})$. The last component of $\hat{x}(k \mid k)$ is the desired estimate. Assuming $P^{-1}(k \mid k-1)$ exists, a useful alternative form for (2b) is

$$P^{-1}(k \mid k) = P^{-1}(k \mid k-1) + H^T(k)H(k)r_k^{-1}$$

We subsequently take the point of view that the model is not necessarily correct and/or the initial conditions $\hat{x}(0 \mid 0)$ and $P(0 \mid -1) > 0$ are arbitrary, and investigate the stability of the KF (2). We want to demonstrate the uniform asymptotic stability (in the large) of the KF (2) or more precisely of the free equation

$$\xi(k) = (I - K(k)H(k))A\xi(k-1) \tag{3}$$

(see [5] for basic definitions and results on stability). It is known that for time-varying linear systems uniform asymptotic stability implies exponential stability and bounded input bounded state stability. Furthermore, for random inputs, bounded input bounded state stability holds in the mean-square sense.

Stability results for the KF (2) which follow directly from standard theorems in the

literature can be summarized as follows. First, in the case where the system (1) is time-invariant, i.e., $c_i(k) = c_i$, $q_k = q$, $r_k = r$, the KF (2) is asymptotically time invariant and asymptotically stable since the system transition matrix A is stable [6]. For the case of general time-varying systems, stability results for KFs can be found in [7,8]. These results assert that if the system transition matrix is invertible and certain uniform observability and controllability conditions are satisfied, then uniform asymptotic stability is obtained. However, in our case the system transition matrix A is singular and the observability condition generally does not hold.

Below we state a result valid for the KF (2) without assuming the system (1) is time-invariant or even asymptotically time-invariant. Let

$$L(k) = \max\{i \le K : |c_i(k)| > 0\},$$

the channel length at time k. In the sequel we make the following assumptions

(A1) $\inf_k r_k > 0$, $\sup_k r_k < \infty$; $\inf_k q_k > 0$, $\sup_k q_k < \infty$

$\sup_k |c_i(k)| < \infty$, $\quad i = 0,...,K$

(A2) $\lim_{k\to\infty} L(k) = L$ (i.e., $L(k) = L$ for k large enough)

Here is our main result.

Theorem The KF (2) is uniformly asymptotically stable (in the large) if $\liminf_{k\to\infty} |c_L(k)| > 0$

The proof of the Theorem is sketched in Section 3 (full details will appear elsewhere). Since K(k) is bounded[2], if y_k is mean-square bounded then the uniform asymptotic stability implies that $\hat{x}(k \mid k)$ is mean-square bounded. We remark that if the model (1) is correct then the fact that A is stable implies x(k) and hence $\hat{x}(k \mid k)$ are trivially mean-square bounded. Our result shows that $\hat{x}(k \mid k)$ is mean-square bounded even if the model is incorrect (the usual case!) so long as y_k is mean-square bounded.

3. Proofs

The proof of uniform asymptotic stability for KF's for general time-varying statespace models [7,8] proceeds by obtaining upper and lower bounds on P(k|k). Uniform complete observability and complete controllability are assumed and used to obtain the upper and lower bounds, respectively. For the lower bound, an argument based on defining a backward system is used. The invertibility of the state transition matrix is used in an essential way here. The bounds on P(k|k) are then used with the assumed uniform complete observability to produce a Liapunov function and hence obtain uniform asymptotic stability. The invertibility of the state transition matrix is again used in this last step in a nontrivial way.

Our approach to the proof of the Theorem is to develop upper and lower bounds on P(k|k) which do not rely on observability or controllability properties (directly). We then

---

[2] i.e., bounded in norm; without loss of generality we assume that all norms $\| \ \|$ are Euclidean norms

modify the proof in [7] which provides a Liapunov function (this part does require a certain notion of observability for a reduced system). Problems with the singularity of the state transition matrix A are avoided by using certain matrix inequalities.

We start by giving a proposition with the key matrix inequalities.

Proposition Let P be an nxn positive definite matrix, and $Q = P^{-1}$. Define the following partitions of P and Q:

$$P = \left[\begin{array}{c|c} P_0 & p_0 \\ \hline p_0^T & p_{nn} \end{array}\right], \qquad Q = \left[\begin{array}{c|c} Q_0 & q_0 \\ \hline q_0^T & q_{nn} \end{array}\right],$$

where P_0 and Q_0 are (n-1)x(n-1) (positive definite) submatrices. Then

(a) $P_0^{-1} \le Q_0$

(b) $$\left[\begin{array}{c|c} P_0^{-1} & 0 \\ \hline 0^T & 0 \end{array}\right] \le Q$$

Proof: (b) clearly implies (a), and (b) follows from the standard block matrix inversion Lemma (we omit the details) •

Next, we give a lemma asserting upper and lower bounds on $P(k|k)$.

Lemma 1 There exists constants $\alpha_1, \alpha_2 > 0$ such that

$$\alpha_1 I \le P(k|k) \le \alpha_2 I, \qquad k \ge 0.$$

Proof: It is easy to show that

$$P(k|k) \le AP(k-1|k-1)A^T + GG^T q_k$$

The upper bound follows from this. To establish the lower bound, we first show by induction that $P(k|k) > 0$ for all k. Using Proposition (a) we can then show

$$P^{-1}(k|k) \le AP^{-1}(k-1|k-1)A^T + GG^T q_k^{-1} + H^T(k)H(k)r_k^{-1}$$

The lower bound follows from this. •

Remark: Clearly, we also have $\alpha_3 I \le P(k|k-1) \le \alpha_4 I$ for some $\alpha_3, \alpha_4 > 0$.

Next, we explore the observability issue. Let $\Phi(k) \in \mathbb{R}^{nxn}$, for k = 0,1,..., and let $\Phi(k,j) = \Phi(k-1)\cdots\Phi(j)$ for $k \ge j$. We define $(\Phi(\cdot), \Gamma(\cdot))$ to be *uniformly completely observable* if there exists a constant $\alpha > 0$ and integers M,N such that

$$\sum_{i=k-M}^{k} \Phi^T(i,k-M)\Gamma^T(i)\Gamma(i)\Phi(i,k-M) \ge \alpha I, \qquad k \ge N.$$

(This is a slightly different definition then is given in [8]).

We will abuse notation and let A denote a (downward shift) matrix of appropriate size (without explicitly specifying its size). Now in view of the shifting property of A we have that $(A, H(\cdot))$ is uniformly completely observable only if K = L, i.e., only if lags larger than the limiting channel length are not allowed (and we want to allow such lags). Let

$$C(k) = [c_0(k) \cdots c_L(k)], \qquad k = 0,1,....$$

It turns out that the uniform complete observability of $(A, C(\cdot))$ and not $(A, H(\cdot))$ is critical. In the following lemma we give a characterization.

Lemma 2 $(A, C(\cdot))$ is uniformly completely observable iff $\liminf_{k\to\infty} |c_L(k)| > 0$

Proof: omitted •

Proof of Theorem: It will be shown that

$$V(\xi(k), k) = \xi(k)^T P^{-1}(k \mid k)\xi(k)$$

is a Liapunov function for the free system (3), i.e., there exists nonegative nondecreasing continuous functions $\varphi_i(\|\xi(k)\|)$, $i = 1,2,3$, with $\varphi_i(0) = 0$, $\varphi_i(\|\xi(k)\|) > 0$ for $\xi(k) \neq 0$, and $\varphi_2(\|\xi(k)\|) \to \infty$ as $\|\xi(k)\| \to \infty$, and an integer N such that

$$\varphi_1(\|\xi(k)\|) \le V(\xi(k), k) \le \varphi_2(\|\xi(k)\|), \qquad k \ge N, \tag{4}$$

$$V(\xi(k), k) - V(\xi(k-1), k-1) \le -\varphi_3(\|\xi(k)\|), \qquad k \ge N. \tag{5}$$

(4) follows from Lemma 1. (5) is established as follows.

Following [7] express the free system (3) as

$$\xi(k) = A\xi(k-1) + u(k).$$

(this defines u(k)). Then

$$\xi^T(k)P^{-1}(k \mid k)\xi(k) = \xi^T(k-1)A^T P^{-1}(k \mid k-1)A\xi(k-1) - \xi^T(k)H^T(k)r_k^{-1}H(k)\xi(k)$$
$$-u^T(k)P^{-1}(k \mid k-1)u(k)$$

Using Proposition (b) we can show that

$$A^T P^{-1}(k \mid k-1)A \le P^{-1}(k-1 \mid k-1)$$

Therefore

$$V(\xi(k), k) - V(\xi(k-1), k-1) \le -\xi^T(k)H^T(k)r_k^{-1}H(k)\xi(k) - u^T(k)P^{-1}(k \mid k-1)u(k)$$

and so

$$V(\xi(k), k) - V(\xi(k-K-1), k-K-1) \le -\sum_{i=k-K}^{k}\left[\xi^T(i)H^T(i)r_i^{-1}H(i)\xi(i) + u^T(i)P^{-1}(i \mid i-1)u(i)\right]$$
$$=: -J$$

Observe that J is a function of ξ(k-K-1) and u(i), $i = k-K,\ldots,k$, once $\xi(i)$, $i = k-K,\ldots,k$, are eliminated in favor of these variables. Let J^* denote the minimum value of J over u(i), $i = k-K,\ldots,k$, with ξ(k-K-1) fixed. This is the same minimum problem considered in [7]. It follows similarly as in [7] that there exists $\beta_1 > 0$ such that

$$J^* \ge \beta_1 \xi^T(k-K-1)A^T\left[\sum_{i=k-K}^{k}(A^T)^{i-k+K}H^T(i)H(i)A^{i-k+K}\right]A\xi(k-K-1).$$

Let $\xi(k) = [\xi_1^T(k) : \xi_2^T(k)]^T$ where $\xi_1(k), \xi_2(k)$ have dimension $L+1$ and $K-L$, respectively. By assumption, there exists an integer N_1 such that the channel length $L(k) = L$ for $k \ge N_1$. Hence, in view of the fact that for $k \ge N_1$ H(k) has its last K - L components zero,

$$J^* \ge \beta_1 \xi_1^T(k-K-1)A^T\left[\sum_{i=k-K}^{k-K+L}(A^T)^{i-k+K}C^T(i)C(i)A^{i-k+K}\right]A\xi_1(k-K-1)$$

for $k \ge N_1 + K$. Now by Lemma 2 $(A, C(\cdot))$ is uniformly completely observable. Hence,

there exists an integer N_2 and $\beta_2 > 0$ such that

$$J^* \geq \beta_2 \|A\xi_1(k-K-1)\|^2, \qquad k \geq N_2.$$

At this point we make the following

Claim There exists an integer N_3 and $\beta_3 > 0$ such that

$$\|A\xi_1(k-K-1)\| \geq \beta_3 \|\xi(k)\|, \quad k \geq N_3.$$

Suppose the Claim is true. Then

$$V(\xi(k),k) - V(\xi(k-K-1),k-K-1) \leq -J^* \leq -\beta_2\beta_3^2\|\xi(k)\|^2, \qquad k \geq N_2 \vee N_3.$$

This implies (5) and hence the Theorem. The Claim is proved by expressing the free system (3) in the form

$$\xi_1(k) = \Phi_1(k)A\xi_1(k-1)$$

$$\xi_2(k) = \Phi_2(k)\xi_1(k-1) + A\xi_2(k-1), \qquad k \geq N_1,$$

and using some elementary bounding arguments. •

4. References

[1] J.G. Proakis, *Digital Communications* . New York: McGraw-Hill, 1989.

[2] A.T. Walden, "Non-Gaussian reflectivity, entropy and deconvolution," *Geophysics*, vol. 50, no. 12, pp. 2862-2888, 1985.

[3] J.V. Candy, *Signal Processing: The Model-Based Approach* . New York: McGraw-Hill, 1986.

[4] G.K. Lee, S.B. Gelfand, and M.P. Fitz, "Bayesian techniques for known channel deconvolution," *Proc. 31st Annual Allerton Conf. Commun., Contr., and Computing*, Monticello, Illinois, pp. 977-986, Sept. 1993 (also accepted pending revision in *IEEE Trans. Commun.*)

[5] R.E. Kalman and J.E. Bertram, "Control system analysis and design via the second method of Liapunov, II discrete-time systems," *Trans. ASME, J. Basic Engrg.*, pp. 394-400, 1960.

[6] B.D.O. Anderson and J.B. Moore, *Optimal Filtering* . New Jersey: Prentice-Hall, 1979.

[7] J.J. Deyst and C.F. Price, "Conditions for asymptotic stability of the discrete minimum variance linear estimator," *IEEE Trans. Automat. Contr.*, vol. AC-13, pp. 702-707, 1968.

[8] A.H. Jazwinski, *Stochastic Processes and Filtering Theory* . New York: Academic Press, 1970.

A NOVEL ADAPTIVE NOTCH FILTER FOR HOWLING CANCELLATION

M. Tahernezhadi, L. Liu, and G. Miller
Department of Electrical Engineering
Northern Illinois University
DeKalb, IL 60115

ABSTRACT A new auto-regressive based (AR) adaptive notch filtering algorithm for howling control is formulated. The algorithm, being both time and order recursive, consists of a lattice structure followed by an all pole filter. The roots of predictor polynomials, associated with each stage of the lattice filter, are constrained to be on the unit circle and in complex conjugate pairs. The all pole filter, with the last stage output as its input, has all its poles equal to zeros of the last stage predictor polynomial multiplied by a positive number less than unity. Simulation results attest to the algorithm's fast tracking and adequate suppression of howling components.

I. INTRODUCTION

Howling is the presence of slow changing and relatively high amplitude sinusoidal components embedded in broadband speech signal. It comes about as a result of feedback path between the microphone unit(s) and the speaker unit(s), for example, in a handset-free telephone. Adaptive notch filtering [1]-[5] serves as an effective means for howling cancellation. In this paper, a new adaptive howling controller is proposed based on a constrained pole-zero notch filter [1]-[3] in conjunction with a lattice algorithm built around an auto-regressive (AR) based frequency estimation technique [6]. In [6], a symmetric Lanczos algorithm for computing the dominant frequencies of a block of sample points is presented. Central to this frequency estimation algorithm is that the predictor polynomial is constrained to have its roots on the unit circle and in complex conjugate pairs. The objective in this paper is not to estimate the underlying frequencies as was done in [6], but to compute the constrained predictor polynomial in an order and time recursive fashion one sample at a time rather than block processing. As such, the algorithm consists of a lattice filter followed by an all pole filter. The lattice filter at each stage computes the error sequence and the coefficients of the constrained predictor polynomial. Subsequently, the error sequence of the last stage excites an all-pole filter whose poles are equal to the zeros of the last stage predictor polynomial multiplied by a positive constant less than unity.

This paper is organized as follows. In Section II, a lattice structure is presented to compute the constrained predictor polynomials along with their associated error sequences in a time and order recursive manner. Section III presents the constrained zero-pole adaptive notch filter. Numerical results for howling control application are presented in Section IV. Finally, conclusions are presented in section V.

II. A Constrained Lattice Structure

A signal x(n) consisting of p dominant sinusoidal components can be described by a 2p AR model as

$$x(n) = -\sum_{k=1}^{2p} a_k x(n-k) + q_p(n) \tag{1}$$

with the constraint that the predictor polynomial $A_p(Z)$ given as

$$A_p(Z) = 1 + \sum_{k=1}^{2p} a_k Z^{-k}, \tag{2}$$

having all its roots on the unit circle; Z^{-1} denotes the unit delay operator. The estimation error, $q_p(n)$, is given by

$$q_p(n) = A_p(Z)x(n) \tag{3}$$

The predictor polynomial $A_p(Z)$, owing to the above-stated constraint, can be factorized as

$$A_p(Z) = Z^{-p} \prod_{k=1}^{p} (\Omega - 2\cos(w_k))$$

$$= Z^{-p} B_p(\Omega) \tag{4}$$

where w_k's are the frequencies of underlying sinusoidal components in x(n), $\Omega = Z + Z^{-1}$ is a symmetric shift operator [6], and $B_p(\Omega)$ is a p-th order polynomial with roots in the interval [-2,2].

At this point let us define the inner product between x(n) and x(n+k) as

$$< x(n), x(n+k) > = E(x(n)x(n+k))$$

$$= r_k \tag{5}$$

where E denotes the expectation operator, and r_k denotes the k-th correlation coefficient assuming a stationary sequence x(n). The estimation error power, ε_p, can be obtained as

$$\varepsilon_p = < q_p(n), q_p(n) >$$

$$= \| Z^{-p} B_p(\Omega) x(n) \|^2$$

$$= \| B_p(\Omega) x(n) \|^2 \tag{6}$$

Thus, the objective is to find that p-th order monic polynomial, $B_p(\Omega)$, that minimizes (6) with respect to the inner product defined in (5). It can be shown [7] that when $B_p(\Omega)$ is a p-th order monic orthogonal polynomial, (6) is minimized. Furthermore, the orthogonal polynomials of order j+1, j, and j-1 obey the triple recursion relation given by [6]

$$B_{j+1}(\Omega) = (\Omega - \alpha_j) B_j(\Omega) - \beta_j B_{j-1}(\Omega), \tag{7}$$

where α_j and β_j are readily obtained from the orthogonality condition as

$$\alpha_j = \frac{< \Omega B_j(\Omega) x(n), B_j(\Omega) x(n) >}{\varepsilon_j}, \tag{8a}$$

$$\beta_j = \frac{< \Omega B_j(\Omega) x(n), B_{j-1}(\Omega) x(n) >}{\varepsilon_{j-1}}. \tag{8b}$$

In addition, it directly follows from (3), (4) and (7) that the error $q_{j+1}(n)$, due to a 2(j+1) constrained AR model, obeys the following lattice recursion

$$q_{j+1}(n) = q_j(n) + q_j(n-2) - \alpha_j q_j(n-1)$$

$$- \beta_j q_{j-1}(n-2) \tag{9}$$

Moreover, it can be shown that α_j and β_j, given in (8), can be expressed in terms of the error sequences q_j and q_{j-1} as

$$\alpha_j = \frac{< q_j(n) + q_j(n-2), q_j(n-1) >}{\| q_j(n-1) \|^2} \tag{10a}$$

$$\beta_j = \frac{< q_j(n) + q_j(n-2), q_{j-1}(n-2) >}{\| q_{j-1}(n-2) \|^2} \tag{10b}$$

The numerators and denominators in (10a) and (10b) are be updated recursively in time according to

$$u_j(n) = \lambda u_j(n-1) + [q_j(n) + q_j(n-2)] q_j(n-1) \tag{11a}$$

$$v_j(n) = \lambda v_j(n-1) + q_j^2(n-1) \tag{11b}$$

$$w_j(n) = \lambda w_j(n-1) + [q_j(n) + q_j(n-2)] q_{j-1}(n-2) \tag{11c}$$

where λ is a forgetting factor. Then,

$$\alpha_j(n) = \frac{u_j(n)}{v_j(n)} \tag{12a}$$

$$\beta_j(n) = \frac{w_j(n)}{v_{j-1}(n)} \tag{12b}$$

Also, by utilizing (4) in (7), one obtains the following lattice recursive relation for the coefficients of the polynomial $A_j(Z)$

$$a_{k,j+1}(n) = a_{k,j}(n) + a_{k-2,j}(n) - \alpha_j(n) a_{k-1,j}(n) - \beta_j(n) a_{k-2,j-1}(n) \tag{13}$$

with initial conditions given as

$$a_{0,j}(n) = 1 \qquad j = 0, \ \ldots, \ p, \quad n \geq 0 \tag{14a}$$

$$a_{k,j}(n) = 0 \qquad k, j < 0 \tag{14b}$$

where $a_{k,j}(n)$ denotes the k-th coefficient of the polynomial $A_j(Z)$ at time n.

Recapitulating, (9) and (11)-(14) form a time and order recursive algorithm to compute the coefficients of the constraint polynomial, $A_j(Z)$, and the error sequence, $q_j(n)$, when an input signal x(n) consisting of p dominant sinusoidal components is presented. In what follows, we will employ this algorithm in order to construct an adaptive notch filter as to suppress the dominant sinusoidal components in the signal x(n).

III ADAPTIVE NOTCH FILTER WITH CONSTRAINED POLES AND ZEROS

We consider a class of IIR notch filters with constrained pole and zeros [1]-[2]. The constraint is to have zeros on the unit circle and poles equal to the zeros multiplied by a factor r, where r is a positive number less than unity. This constraint provides a good approximation to an ideal notch filter. The transfer function of the notch filter is given by

$$N(z) = \frac{A(Z)}{A(Z/r)} \tag{15}$$

where A(Z) is a polynomial with roots on the unit circle. The pole radius, r, determines the bandwidth of the filter and should be fairly close to unity in order to well approximate the ideal notch filter. Since $A_p(Z)$ has its zeros on the unit circle, therefore it can be directly used in (15). Moreover, when implemented with the proposed lattice structure, the output of notch filter is the convolution of the lattice last stage error sequence, $q_p(n)$, with an IIR filter whose transfer function is given by

$$W(Z) = \frac{1}{A_p(Z/r)} \tag{16}$$

The k-th parameters of W(Z) is equal to k-th the coefficient of $A_p(Z)$ multiplied by r^k.

III APPLICATION TO HOWLING CONTROL

Section II and III formulated an adaptive notch filter algorithm for cancellation of dominant sinusoidal components in the input signal, x(n). For howling control application, the input signal consist of speech plus howling components, whose number is unknown. Thus, we need to address two problems: (i) how to determine the number of the howling components, p; and (ii) how to distinguish dominant frequencies in speech from those in howling.

The number of howling components can be determined by monitoring the error power at the output of each stage. It follows from (7) and (8b) that the error power of stage j is given by

$$\begin{aligned}\varepsilon_{j+1} &= < B_{j+1}(\Omega)x(n), B_{j+1}(\Omega)x(n) > \\ &= < \Omega B_j(\Omega)x(n), B_{j+1}(\Omega)x(n) > \\ &= \beta_{j+1}\varepsilon_j\end{aligned} \tag{17}$$

It is instructive to point out that in (17) we have also exploited the relationship that $< \Omega x(n), x(k) > = < x(n), \Omega x(k) >$. Thus, in turn, $\beta_j(n)$ given in (12b) can be monitored and when its value falls below a certain threshold, the order updation is ceased.

In order to distinguish the dominant frequencies in speech from those in howling, we exploit the fact that howling frequencies are moving much slower across the audio spectrum than those of speech. Therefore, in what follows, we propose a parameter, $\Delta(n)$ that flags the presence of howling. This parameter is created around the fact that, owing to the triple recursion relationship given in (7), changes in zeros of orthogonal polynomials and in turn in the frequencies lead to variations in paramters α_j's . These variations are slow for howling frequencies and fast for speech frequencies. In particular, $\Delta(n)$ is created to track the combined changes in $\alpha_o(n)$ and $\alpha_1(n)$ as

$$\Delta(n) = \lambda\Delta(n-1) + \Delta_o(n) + \Delta_1(n) \tag{18a}$$

where

$$\Delta_o(n) = |\alpha_o(n) - \alpha_o(n-\delta)| \tag{18b}$$

$$\Delta_1(n) = |\alpha_1(n) - \alpha_1(n-\delta)| \tag{18c}$$

and δ is a delay parameter. For example, the simulation results of Fig. 1 demonstrate the behavior of $\Delta(n)$ for a pure speech signal and howling+speech signal when the forgetting factor is $\lambda = 0.98$ and $\delta = 15$. It is seen that while $\Delta(n)$ remains relatively small for howling+speech signal, it exhibits significant fluctuation for a pure speech signal. Thus, by monitoring the behavior of $\Delta(n)$, the algorithm can detect the presence of howling.

The simulation results shown in Fig. 2 attest to fast convergence behavior of the algorithm. Although the algorithm only computes the constraint predictor polynomial of each lattice stage, in this figure we have plotted the associated frequencies of the constrained predictor polynomial of the last stage. It is seen that convergence for two howling components with normalized frequencies of 0.7 and 1.2 is achieved in about 60 iterations.

V. Conclusions

In this paper, we have formulated an orthogonal polynomial based time and order recursive adaptive notch filter for howling cancellation. Effectively, the algorithm resulted in a constrained pole-zero adaptive notch filter with zeros on the unit circle and poles equal to zeros multiplied by a positive constant less than unity. Simulation results attested to fast tracking and good suppression capability of the proposed algorithm in howling control applications.

REFERENCES

[1] A. Nehorai, "A minimal parameter adaptive notch filter with constrained pole and zeros," *IEEE Trans. Acoust., Speech, and Signal Processing*, ASSP-33, pp. 983-996, April 1985.

[2] T. S. Ng, "Some aspects of an adaptive digital notch filter with constrained poles and zeros," *IEEE Trans. Acoust., Speech, and Signal Processing*, ASSP-35, pp. 158-161, Feb. 1987.

[3] P. Stoica, and A. Nehorai, "Performance analysis of an adaptive notch filter with constrained poles and zeros," *IEEE Trans. Acoust., Speech, and Signal Processing*, ASSP-36, pp. 911-919, June 1988.

[4] B. D. V. Rao, and S. Y. Kung, "Adaptive notch filtering for the retrieval of sinusoids in noise," *IEEE Trans. Acoust., Speech, and Signal Processing*, ASSP-32, pp. 791-802, April 1984.

[5] P. A. Regalia, "An improved lattice-based adaptive IIR notch filter," *IEEE Trans. Signal Processing*, SP-39, pp. 2124-2128, Sep. 1991.

[6] G. Cybenko, "Fast approximation of dominant harmonics," *SIAM J. Sci. Stat. Comput.*, vol. 5, no. 2, pp. 317-331, June 1984.

[7] N. Grenander and G. Szego, *Toeplitz forms and their application*, Berkeley, CA, University Calif Press ,1958.

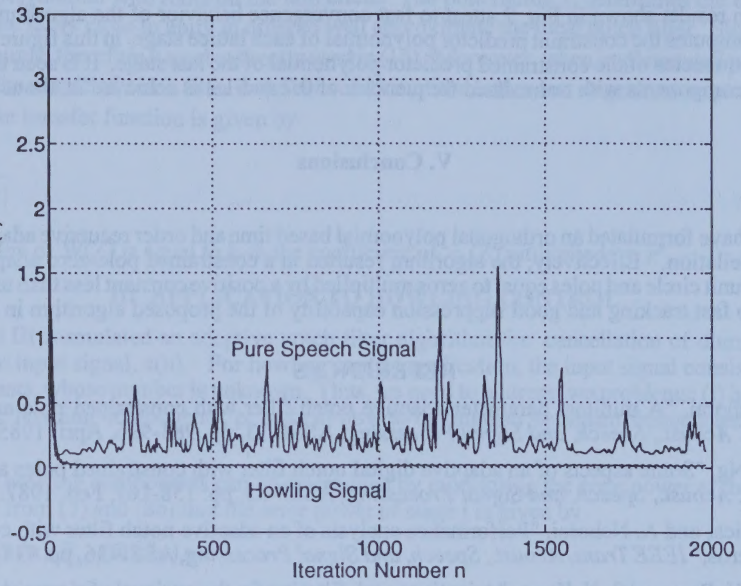


FIG. 1. BEHAVIOR OF $\Delta(n)$ IN THE PRESENCE OF HOWLING+SPEECH AND PURE SPEECH, $\lambda = 0.98$ and $\delta = 15$.

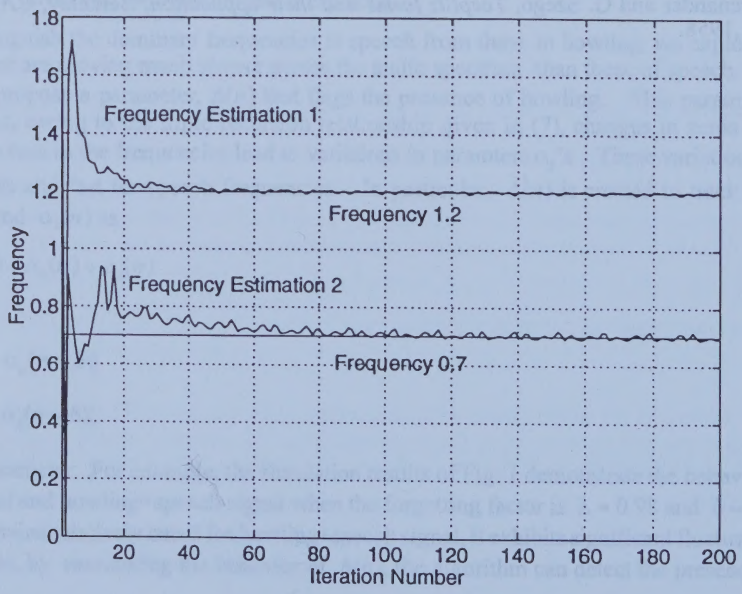


FIG. 2. CONVERGENCE OF FREQUENCY ESTIMATES OF THE PROPOSED ALGORITHM.

A Low-Complexity Pole-Zero MMSE Equalizer for ML Receivers

Naofal Al-Dhahir *and John Cioffi

Information Systems Laboratory

Stanford University, Stanford CA 94305

Abstract

We present a *unified* approach for computing the optimum settings of a length-N_f input-aided equalizer that minimizes the mean square error between the equalized channel output and a target response of a given length N_b. The presented approach offers more insight into the problem, encompasses a multitude of equalization structures, easily accommodates correlation in the input and noise sequences, leads to significant computational savings, and allows us to analyze a variety of constraints on the target response besides the unit-tap constraint assumed in [6].

Furthermore, we show that, under the assumed constraint of finite-length filters, the relative delay between the equalizer and the target response plays a crucial role in optimizing performance. We describe a *new* characterization of the delay and show how to optimize it.

Finally, we derive reduced-parameter pole-zero models of the equalizer that achieve the high performance of a long FIR equalizer at a much lower cost of implementation.

1. Introduction

There was a flurry of research activity during the Seventies (see e.g. [9, 16, 13, 12]) to combine equalization with maximum likelihood sequence estimation (MLSE) to achieve higher data rates on bandlimited noisy linear channels. The function of the equalizer is to "condition" the channel to reduce the channel impulse response length, denoted henceforth by $(\nu + 1)$. This, in turn, allows for a practical use of the high-performance Viterbi algorithm (VA) [11] whose computational complexity increases exponentially with the channel memory (ν).

Several criteria were considered for choosing the reduced-length target impulse response (TIR) that results from equalizing the original channel impulse response. Qureshi and Newhall [16] chose a truncated version of the original channel impulse response, while Falconer and Magee [9] proposed to minimize the mean square error (MSE) between the channel-equalizer combination and the TIR, subject to a unit-energy constraint on the latter. In [13], Messerschmitt suggested a monicity constraint on the TIR and used the whitened-matched-filter [11] as the equalizer structure. To the best of our knowledge, in all of the previous work, with the exception of [9], an infinite-length equalizer was assumed. This is an idealistic assumption since all practical implementations require a finite number of filter coefficients.

More recently in [6, 7], there has been a renewed interest in finite-complexity equalizers in conjunction with another form of ML receivers, namely multicarrier demodulators. In multicarrier modulation (MCM), the channel spectrum is partitioned into a large number of orthogonal, parallel, and (approximately) memoryless subchannels. Several forms of MCM have been developed depending on the choice of the modulating and demodulating vectors used. In a particular cost-effective implementation of MCM known as the discrete multitone transceiver (DMT)

*Naofal Al-Dhahir is now with GE Corporate R &D, Schenectady, NY 12301.

[8], the discrete Fourier transform (DFT) basis vectors are used as the demodulating vectors and are implemented using the computationally–efficient FFT algorithm. To prevent inter–block interference between successive input blocks, each input block of length N is appended by ν symbols (a cyclic prefix whose length is equal to the channel memory). This results in a data rate reduction by a factor of $\frac{\nu}{N+\nu}$. In situations where severe ISI is present (i.e., ν is large) and where end–to–end delay restrictions prohibit the use of a long block length N, the data rate reduction due to the cyclic prefix becomes quite significant. The use of a finite–length mean–square–error equalizer to reduce the size of the cyclic prefix was proposed and analyzed in [6]. More recently, a *new* equalization criterion based on maximizing the *geometric SNR* was proposed and analyzed in [4].

Our objective in this paper is to generalize the finite–length MSE equalization structure of [9, 6] in several directions. First, we shall show how the unit–energy constraint (UEC) of [9] and the unit–tap constraint (UTC) of [6] can be derived under a single unified framework that also lends itself to an easy treatment of other constraints. Second, we extend the results of [9, 6] to include correlation in the input and noise sequences. This allows us to consider the effects of transmit filter shaping, coding, and non–white impairments such as crosstalk and filtered noise. In addition, we shall assume an input–aided equalization structure [14] in which perfect knowledge of a finite number of past and future data symbols is assumed. Decision–aided equalizers are a subclass of this general structure when the used decisions are assumed to be correct. We shall show in this paper how a host of equalization structures follow as special cases of our general framework.

Moreover, in contrast to the adaptive methods suggested in [9] for computing the optimum equalizer and TIR settings, we shall focus on non–adaptive channel–estimate–based methods. Our main motivation being the slow convergence of traditional time–domain LMS adaptation algorithms in this application [1](as verified by the simulation results of [12]), coupled with recent results (see [17] and the references therein) that showed undeniable performance advantages in computing the equalizer settings based on updated channel and noise estimates as opposed to updating the equalizer coefficients directly. Among our contributions in this paper is an efficient procedure (that requires much less computations than the one described in [6]) for computing the optimum equalizer and TIR settings anew from updated channel and noise estimates. Furthermore, as pointed out in [6], when finite–length filters are used, the relative delay between the equalizer and TIR becomes of paramount importance. Optimization of the delay was not performed in [9]. We shall characterize the delay in terms of the starting index of a submatrix of a channel–and–noise–dependent matrix. This characterization allows us to devise a search procedure for the optimum delay that is much more efficient than the one presented in [6]. Finally, we shall derive pole–zero models of the equalizer that will be demonstrated to achieve high performance at substantial reductions in computational and implementational complexities.

2. Model and Assumptions

2.1 A Unified Approach

The block diagram of a MMSE input–aided equalizer (IAE) is shown in Figure 1. The channel pulse response is assumed to be linear time–invariant and consists of $(\nu+1)$, generally complex, taps $\{\mathbf{h}_0, \mathbf{h}_1, \cdots, \mathbf{h}_\nu\}$. The coefficient $\mathbf{h}_m$ $(0 \leq m \leq \nu)$ is an l–dimensional column vector that contains the l samples of the m^{th} symbol period and l is denoted as the oversampling factor. This representation accommodates the general case of fractionally–spaced equalizers that will be assumed throughout this paper, unless otherwise stated.

We assume that both the input sequence $\{x_k\}$ and the noise sequence $\{n_k\}$ consist of complex, zero–mean, and correlated symbols. Furthermore, the noise symbols are assumed to be *independent* of the input symbols.

[1]This has prompted the authors in [7] to consider frequency domain FFT–based adaptation algorithms instead.

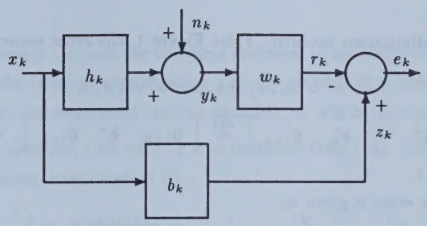


Figure 1: Block diagram of the FIR MMSE–IAE

Over a block of N_f output symbols (each comprised of l samples), the input–output relationship can be cast in matrix form as follows :

$$\mathbf{y} = \begin{bmatrix} \mathbf{h}_0 & \mathbf{h}_1 & \cdots & \mathbf{h}_\nu & 0 & \cdots & 0 \\ 0 & \mathbf{h}_0 & \mathbf{h}_1 & \cdots & \mathbf{h}_\nu & 0 & \cdots \\ \vdots & & & & & & \vdots \\ 0 & \cdots & 0 & \mathbf{h}_0 & \mathbf{h}_1 & \cdots & \mathbf{h}_\nu \end{bmatrix} \mathbf{x} + \mathbf{n} \,,$$

where,

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_{k+N_f-1} \\ \mathbf{y}_{k+N_f-2} \\ \vdots \\ \mathbf{y}_k \end{bmatrix} ; \mathbf{x} = \begin{bmatrix} x_{k+N_f-1} \\ x_{k+N_f-2} \\ \vdots \\ x_{k-\nu} \end{bmatrix} \mathbf{n} = \begin{bmatrix} \mathbf{n}_{k+N_f-1} \\ \mathbf{n}_{k+N_f-2} \\ \vdots \\ \mathbf{n}_k \end{bmatrix} .$$

Expressed more compactly :

$$\mathbf{y}_{k+N_f-1:k} = \mathbf{H}\mathbf{x}_{k+N_f-1:k-\nu} + \mathbf{n}_{k+N_f-1:k} \,. \tag{1}$$

The $(N_f+\nu)$–dimensional input correlation matrix is defined by $\mathbf{R}_{xx} \stackrel{def}{=} E[\mathbf{x}_{k+N_f-1:k-\nu}\mathbf{x}^*_{k+N_f-1:k-\nu}]$, and the (lN_f)–dimensional noise correlation matrix is defined by $\mathbf{R}_{nn} \stackrel{def}{=} E[\mathbf{n}_{k+N_f-1:k}\mathbf{n}^*_{k+N_f-1:k}]$. Both $\mathbf{R}_{xx}$ and $\mathbf{R}_{nn}$ are assumed to be positive–definite correlation matrices and $(.)^*$ denotes the conjugate transpose operation.

Then, the input–output cross–correlation and the output auto–correlation matrices are given by :

$$\begin{aligned} \mathbf{R}_{xy} &\stackrel{def}{=} E[\mathbf{x}_{k+N_f-1:k-\nu}\mathbf{y}^*_{k+N_f-1:k}] \\ &= \mathbf{R}_{xx}\mathbf{H}^* \qquad (2) \\ \mathbf{R}_{yy} &\stackrel{def}{=} E[\mathbf{y}_{k+N_f-1:k}\mathbf{y}^*_{k+N_f-1:k}] \\ &= \mathbf{H}\mathbf{R}_{xx}\mathbf{H}^* + \mathbf{R}_{nn} \,. \qquad (3) \end{aligned}$$

Given the length, (N_b+1), of the TIR b_k, our objective is to compute the (lN_f) coefficients of the filter w_k and the (N_b+1) coefficients of the filter b_k such that the mean square of the error sequence, $e_k \stackrel{def}{=} z_k - r_k$, is minimized. Note that, unlike the FIR MMSE–DFE case [3], the filter b_k is *not restricted to be causal or monic*. Therefore, two extra parameters can be introduced to further improve the performance over that of the FIR MMSE–DFE. First, a relative delay Δ $(0 \le \Delta \le N_f + \nu - N_b - 1)$ between the equalizer response $w(D) \stackrel{def}{=} \mathbf{w}_0 + \mathbf{w}_1 D + \cdots + \mathbf{w}_{N_f-1} D^{N_f-1}$ and the TIR $b(D) \stackrel{def}{=} b_0 D^\Delta + b_1 D^{\Delta+1} + \cdots b_{N_b} D^{N_b+\Delta}$. Second, an index parameter i $(0 \le i \le N_b)$ that sets the i^{th} tap of $b(D)$ to unity.

Now, we go back to our optimization problem. From Figure 1, the error sequence, e_k, can be expressed as :

$$e_k \stackrel{def}{=} \tilde{\mathbf{b}}^* \mathbf{x}_{k+N_f-1:k-\nu} - \mathbf{w}^* \mathbf{y}_{k+N_f-1:k} , \tag{4}$$

where $\tilde{\mathbf{b}}^* \stackrel{def}{=} \left[\begin{array}{ccccccc} \mathbf{0}_{1\times\Delta} & b_0^* & b_1^* & \cdots & b_{N_b}^* & \mathbf{0}_{1\times s} \end{array} \right] \stackrel{def}{=} \left[\begin{array}{ccc} \mathbf{0}_{1\times\Delta} & \mathbf{b}^* & \mathbf{0}_{1\times s} \end{array} \right]$, $\mathbf{w}^* \stackrel{def}{=} \left[\begin{array}{ccc} \mathbf{w}_0^* & \cdots & \mathbf{w}_{N_f-1}^* \end{array} \right]$, and $s \stackrel{def}{=} N_f + \nu - \Delta - N_b - 1$.

Therefore, the mean square error is given by:

$$MSE = \tilde{\mathbf{b}}^* \mathbf{R}_{xx} \tilde{\mathbf{b}} - \tilde{\mathbf{b}}^* \mathbf{R}_{xy} \mathbf{w} - \mathbf{w}^* \mathbf{R}_{yx} \tilde{\mathbf{b}} + \mathbf{w}^* \mathbf{R}_{yy} \mathbf{w} . \tag{5}$$

Using the *orthogonality principle* which states that the *optimal* error sequence is uncorrelated with the observed data, we have $E[e_k \mathbf{y}_{k+N_f-1:k}^*] = 0$. Therefore,

$$\tilde{\mathbf{b}}^* \mathbf{R}_{xy} = \mathbf{w}^* \mathbf{R}_{yy} . \tag{6}$$

Combining (5) and (6), we get:

$$MSE = \tilde{\mathbf{b}}^* \mathbf{R}_{x/y}^{\perp} \tilde{\mathbf{b}} , \tag{7}$$

$$\text{where, } \mathbf{R}_{x/y}^{\perp} = \mathbf{R}_{xx} - \mathbf{R}_{xy} \mathbf{R}_{yy}^{-1} \mathbf{R}_{yx} \tag{8}$$

$$= [\mathbf{R}_{xx}^{-1} + \mathbf{H}^* \mathbf{R}_{nn}^{-1} \mathbf{H}]^{-1} . \tag{9}$$

The last line follows from the matrix inversion lemma [2] (assuming that $\mathbf{R}_{xx}$ and $\mathbf{R}_{nn}$ are invertible).

By defining :

$$\mathbf{R}_{\Delta} \stackrel{def}{=} \left[\begin{array}{ccc} \mathbf{0}_{(N_b+1)\times\Delta} & \mathbf{I}_{N_b+1} & \mathbf{0}_{(N_b+1)\times s} \end{array} \right] \mathbf{R}_{x/y}^{\perp} \left[\begin{array}{c} \mathbf{0}_{\Delta\times(N_b+1)} \\ \mathbf{I}_{N_b+1} \\ \mathbf{0}_{s\times(N_b+1)} \end{array} \right] , \tag{10}$$

it can be shown that MSE is minimized, subject to the unit-tap constraint, $\mathbf{b}^* \mathbf{e}_i = 1$ (where $\mathbf{e}_i$ is the i^{th} unit vector), by the following choice for the TIR coefficients :

$$\mathbf{b}_{opt.} = \frac{\mathbf{R}_{\Delta}^{-1} \mathbf{e}_{i_{opt.}}}{\mathbf{R}_{\Delta}^{-1}(i_{opt.}, i_{opt.})} , \tag{11}$$

where $i_{opt.}$ is the index that achieves the minimum mean square error :

$$MMSE^{UTC} = \frac{1}{\mathbf{R}_{\Delta}^{-1}(i,i)} , \tag{12}$$

and is given by :

$$i_{opt.} = argmax_{0\le i\le N_b} \{\mathbf{R}_{\Delta}^{-1}(i,i)\} . \tag{13}$$

The optimum equalizer settings are determined from (6):

$$\mathbf{w}_{opt.}^* = \tilde{\mathbf{b}}_{opt.}^* \mathbf{R}_{xx} \mathbf{H}^* (\mathbf{H} \mathbf{R}_{xx} \mathbf{H}^* + \mathbf{R}_{nn})^{-1} . \tag{14}$$

Remarks

1. The presented approach offers more geometrical insight into the problem since it formulates the MSE as a *quadratic form in the TIR coefficients.*

[2] The matrix inversion lemma states that

$$(\mathbf{A} + \mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{D}\mathbf{A}^{-1}\mathbf{B} + \mathbf{C}^{-1})^{-1}\mathbf{D}\mathbf{A}^{-1}.$$

2. Although our unified approach requires the one-time inversion of an (lN_f)-dimensional matrix to compute $\mathbf{R}_{x/y}^{\perp}$, it turns out that the inverted matrix has a nice structure that is independent of i and Δ and can be exploited to reduce the computational load significantly, as will be explained more in Section 3.

3. Using our approach, the optimum i for each Δ is determined from (13). Once these optimum indices are computed, then the optimum delay is given by :

$$\Delta_{opt.} = argmax_{0 \le \Delta \le N_f + \nu - N_b - 1} \{ \mathbf{R}_{\Delta}^{-1}(i_{opt.}, i_{opt.}) \} .$$

This a new characterization of the optimum delay that results in a search procedure for $\Delta_{opt.}$ which is more efficient than the one described in [6].

4. Finally, the previous approach of [6] works only under the UTC. In some practical situations, other constraints might be of importance also. For instance, a generalization of the constraint $\mathbf{b}^* \mathbf{e}_i = 1$ would be the constraint $\mathbf{b}^* \mathbf{c} = 1$ where $\mathbf{c}$ is an arbitrary $(N_b + 1)$-dimensional *steering* vector. For example, if $\mathbf{c}$ is the all-ones vector, then the TIR is restricted to have a unity DC gain. Yet another example is when $\mathbf{c}$ is the DFT basis vector $\begin{bmatrix} 1 & e^{-j\omega_0} & \cdots & e^{-jN_b\omega_0} \end{bmatrix}$, in which case the TIR is restricted to have a *distortionless* response at a given frequency ω_0. In either case, the optimum TIR is given by $\mathbf{b}_{opt.} = \frac{\mathbf{R}_{\Delta}^{-1}\mathbf{c}}{\mathbf{c}^* \mathbf{R}_{\Delta}^{-1}\mathbf{c}}$, and the minimum achievable MSE is $\frac{1}{\mathbf{c}^* \mathbf{R}_{\Delta}^{-1}\mathbf{c}}$.

2.2 Another Constraint

We can alternatively impose the condition of a *unit-energy target response*, i.e., $\mathbf{b}^* \mathbf{b} = 1$. It turns out that this constraint results in a lower MSE than that of the UTC.

Under the UEC, it can be shown that $\mathbf{b}_{opt.}$ is equal to the eigenvector corresponding to the minimum eigenvalue of $\mathbf{R}_{\Delta}$ [9], call it λ_{min}. Then, the minimum MSE is :

$$MMSE^{UEC} = \lambda_{min.} . \tag{15}$$

Interestingly enough, if we introduce *weighting* to emphasize some components of the TIR more than others, i.e., if we consider the constraint $\mathbf{b}^* \mathbf{G} \mathbf{b} = 1$ (where $\mathbf{G}$ is a positive-definite matrix), then $\mathbf{b}_{opt.}$ becomes a *generalized eigenvector* of $\mathbf{R}_{\Delta}$.

Next, we shall show that $MMSE^{UEC} \le MMSE_{UTC}$. Define the orthogonal eigen-decomposition :

$$\mathbf{R}_{\Delta} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^* . \tag{16}$$

$$\text{Then, } \mathbf{R}_{\Delta}^{-1} = \mathbf{U} \mathbf{\Lambda}^{-1} \mathbf{U}^* \tag{17}$$

$$\begin{aligned} \Rightarrow \mathbf{R}_{\Delta}^{-1}(i,i) &= (\mathbf{e}_i^* \mathbf{U}) \mathbf{\Lambda}^{-1} (\mathbf{U}^* \mathbf{e}_i) \\ &\stackrel{def}{=} \lambda_0^{-1} |u_{i,0}|^2 + \cdots + \lambda_{N_b}^{-1} |u_{i,N_b}|^2 \\ &\le \lambda_{min}^{-1} \\ \Rightarrow MMSE^{UEC} &= \lambda_{min} \\ &\le \frac{1}{\mathbf{R}_{\Delta}^{-1}(i,i)} \\ &= MMSE^{UTC} , \end{aligned}$$

with equality if and only if all the eigenvalues of $\mathbf{R}_{\Delta}$ are equal. This occurs when the channel is memoryless and the input and noise processes are white.

In Figure 2, we have plotted the variation of the output SNR [3] with delay under both constraints with a fixed choice of N_f and N_b. Under the UTC, we also varied the index parameter to examine its effect on performance

[3] In this paper, output SNR is defined as $\frac{\frac{1}{(N_f+\nu)} trace(\mathbf{R}_{xx})}{MMSE} = \frac{S_x}{MMSE}$ (for white input).

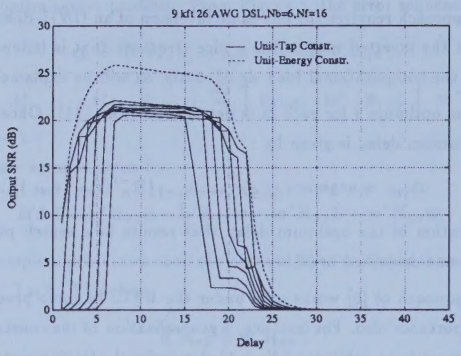


Figure 2: Variation of the output SNR of the MMSE–IAE with delay under both constraints

(with Δ optimized). The simulation parameters are those of a typical high–bit–rate digital subscriber loop (HDSL) environment. More specifically, the channel is assumed to be a 9 kft 26 AWG DSL. The input power level is 17 dBm equally–distributed over the transmission bandwidth. The input signal constellation is 16–QAM and the target bit rate is 800 kbps. The 2–sided AWGN power spectral density is assumed to be -113 dBm/Hz and the standard near–end crosstalk (NEXT) model $|H_x(f)|^2 = k_{NEXT} f^{\frac{3}{2}}$ is adopted with $k_{NEXT} = 10^{-13}$. A symbol–spaced equalizer is assumed. The following conclusions are in order :

1. It is clear that the effect of the delay parameter on output SNR is significant under both constraints. A fixed non–optimized delay could result in an appreciable performance degradation, however, there is typically a range of delay values for which the performance is near–optimum.
2. Under the UTC, the effect of the choice of the index parameter is not crucial, as long as the delay is near its optimum value.
3. The maximum output SNR under the UEC is always higher than its value under the UTC, irrespective of the choices of N_f and N_b, as it was asserted before.

2.3 Some Special and Limiting Cases

The general framework that we have developed in Section 2.1 for analyzing the FIR MMSE IAE includes several well–known equalization structures as special and limiting cases. We explore some of them in this section.

- **The (N_b, N_f) MMSE-DFE**

  Decision feedback equalizers (DFE) are a special case of input–aided equalizers in that the filter $\mathbf{b}$ has access to previous input symbols only, i.e., it is a *causal filter*. Under the further assumption that *previous decisions are correct*, we can use these decisions in place of input symbols. The optimum settings of $\mathbf{b}$ and the resulting MMSE are readily computed from (11) and (12) by substituting $i_{opt} = 0$,[4] giving :

$$\mathbf{b}_{opt.} = \frac{\mathbf{R}_\Delta^{-1}\mathbf{e}_0}{\mathbf{R}_\Delta^{-1}(0,0)} \tag{18}$$

$$MMSE = \frac{1}{\mathbf{R}_\Delta^{-1}(0,0)} . \tag{19}$$

  Again, $\mathbf{w}^*_{opt.}$ is calculated using (14) and the delay parameter Δ is chosen such that (19) is minimized.

[4] In the MMSE–DFE, $\mathbf{b}$ is further restricted to be *monic* since it is a forward prediction filter [10].

- **The FIR MMSE-LE**

  The special case of a MMSE fractionally-spaced linear equalizer (LE) follows immediately by requiring the target response to be memoryless, i.e., $\underline{b} = \mathbf{e}_{\Delta+1}$. Using (7), we have $MMSE = \mathbf{R}^{\perp}_{x/y}(\Delta+1, \Delta+1)$.

- **The (N_b, ∞) MMSE-IAE**

  If we allow the length (N_f) of the symbol-spaced equalizer to become infinite but still require the TIR to be of length (N_b+1), then (14) converges to :

$$w_{opt.}(D) = \frac{R_{xx}(D)h^*(D^{-1})b_{opt.}(D)}{(R_{xx}(D)h(D)h^*(D^{-1}) + R_{nn}(D))} , \tag{20}$$

  where $R_{xx}(D)$ and $R_{nn}(D)$ are the auto-correlation functions of the input and noise sequences, $h(D) = \sum_{i=0}^{\nu} h_i D^i$ and $b_{opt.}(D) = \sum_{i=0}^{N_b} b_i D^i$.

  Moreover, it was shown in [9] that the MSE in this case can be expressed as follows :

$$MSE = \mathbf{b}^* \mathbf{R}^{\infty}_{\Delta} \mathbf{b} , \tag{21}$$

  where $\mathbf{R}^{\infty}_{\Delta}$ is an $(N_b+1) \times (N_b+1)$ *Toeplitz* matrix whose (m,n) element is given by :

$$\mathbf{R}^{\infty}_{\Delta}(m,n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{e^{j\Theta(m-n)}}{\left(\frac{h(e^{j\Theta})h^*(e^{-j\Theta})}{R_{nn}(e^{j\Theta})} + \frac{1}{R_{xx}(e^{j\Theta})}\right)} d\Theta . \tag{22}$$

  The optimum TIR, subject to either the UTC or the UEC, can be determined as it was shown in Section 2 with $\mathbf{R}^{\infty}_{\Delta}$ replacing $\mathbf{R}_{\Delta}$.

  Note that because the equalizer is of infinite length, the delay parameter Δ can be fixed in this case, without loss of optimality. This also follows from the Toeplitz (i.e., time-invariant) structure of $\mathbf{R}^{\infty}_{\Delta}$, unlike the case of finite N_f where $\mathbf{R}_{\Delta}$ is generally non-Toeplitz which makes the optimization of Δ very important.

3. Fast Computation Methods

In this section, we shall enumerate the computational tasks needed to compute $\mathbf{b}_{opt.}$ and $\mathbf{w}^*_{opt.}$ and show how to perform them efficiently. We shall focus on the most common case of white input (i.e., $\mathbf{R}_{xx} = S_x \mathbf{I}_{N_f+\nu}$) and stationary noise (i.e., $\mathbf{R}_{nn}$ is Toeplitz).

1. Computation of $\mathbf{R}^{\perp}_{x/y} = S_x(\mathbf{I}_{N_f+\nu} - \mathbf{H}^*(\mathbf{H}\mathbf{H}^* + \frac{1}{S_x}\mathbf{R}_{nn})^{-1}\mathbf{H})$ requires calculating the inverse of the (lN_f)-dimensional matrix $\mathbf{R} \stackrel{def}{=} \mathbf{H}\mathbf{H}^* + \frac{1}{S_x}\mathbf{R}_{nn}$. This matrix is *block-Toeplitz*. An algorithm that computes $\mathbf{R}^{-1}$ in $O(l^3 N_f^2)$ operations as opposed to $O(l^3 N_f^3)$ operations was developed by Akaike [1]. For the special case of a symbol-spaced equalizer, $\mathbf{R}$ becomes Toeplitz and its inverse can be computed using the Levinson algorithm in $O(N_f^2)$ computations [10].

2. For each value of Δ, $\mathbf{R}_{\Delta}$ is an (N_b+1)-dimensional submatrix of $\mathbf{R}^{\perp}_{x/y}$ that is computed as follows :

$$\mathbf{R}_{\Delta} = S_x(\mathbf{I}_{N_b+1} - \mathbf{H}^*_{\Delta}\mathbf{R}^{-1}\mathbf{H}_{\Delta}) , \tag{23}$$

   where $\mathbf{H}^*_{\Delta} \stackrel{def}{=} \left[\begin{array}{ccc} \mathbf{0}_{(N_b+1)\times\Delta} & \mathbf{I}_{N_b+1} & \mathbf{0}_{(N_b+1)\times s} \end{array}\right] \mathbf{H}^*$ is a submatrix of $\mathbf{H}^*$ of size $(N_b+1) \times (lN_f)$.

3. Computation of $\mathbf{b}_{opt.}$ requires either computing the inverse (under the UTC) or the eigen-decomposition (under the UEC) of $\mathbf{R}_{\Delta}$ which is of $O((N_b+1)^3)$ operations. This is not computationally intense since (N_b+1) is typically small, and can be made even smaller, without sacrificing performance, by increasing N_f (see Example 1 in Section 4).

4. Once $\mathbf{b}_{opt.}$ is determined, $\mathbf{w}^*_{opt.}$ can be calculated from the following two steps (c.f. Equation (14)):

- Compute the (lN_f)-dimensional row vector $\mathbf{b}^*_{opt.}\mathbf{H}^*_\Delta$, which is a *linear combination of the* (N_b+1) *rows of* $\mathbf{H}^*_\Delta$.
- Then, $\mathbf{w}^*_{opt.}$ is equal to the product of the row vector computed in the previous step and $\mathbf{R}^{-1}$ that was computed previously.

To summarize, computation of $\mathbf{b}_{opt.}$ and $\mathbf{w}^*_{opt.}$ requires *one inversion* of a block–Toeplitz matrix (Toeplitz in the symbol–spaced case) and simple multiply–and–add operations that are well–suited for real–time implementation on a digital signal processor.

4. A Reduced-Complexity Implementation

In Figure 3, we have plotted the variation of the maximum output SNR of the MMSE–IAE with N_f and N_b under the UEC. The simulation parameters are identical to those of the HDSL environment that were described in Section 2.2. The lower curve corresponds to the case $N_b = 1$ while the upper curve corresponds to the case $N_b = 16$.

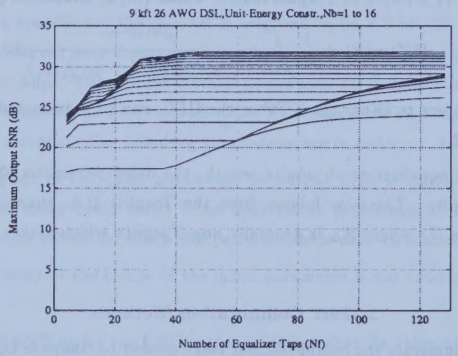


Figure 3: Variation of the maximum output SNR of the MMSE-IAE with N_f and N_b under the UEC

Two main observations can be made based on this plot. First, for intermediate to large values of N_b, increasing the equalizer length N_f results in a gradual increase in the output SNR until it reaches an almost constant level. Further increases in N_f do not result in any discernible SNR improvement. Second, for channels that have a long impulse response (such as the DSL considered in this paper), choosing a short TIR length results in a poor performance *unless the equalizer is made very long.* In fact, a significant portion of the lost output SNR due to a small value of N_b can be recovered by choosing a large value for N_f.

Based on these observations, we propose the following scheme to achieve high performance at a low complexity

- Choose a small value of N_b (that results in a practical implementation of MLSE and a tolerable data rate loss for the chosen block length of the DMT).
- Choose a large enough value of N_f to achieve a satisfactory output SNR level of the MMSE–IAE.
- Convert the long FIR equalizer $\mathbf{w}^*_{opt.}$ to a pole–zero filter with much fewer taps for a reduced–complexity implementation.

The algorithm that we shall use to generate the pole–zero models of $\mathbf{w}^*$ was derived and analyzed in detail in [5]. Here, it suffices to mention that *the algorithm is a 2–channel Levinson algorithm that efficiently generates stable pole–zero models of a given long FIR filter. The number of poles and zeros are not restricted to be equal.*

Model Order	$\hat{w}(D)$	NNTE(dB)
ARMA(5, 1)	$\frac{1.0164-.0346D}{1-.8564D-.0868D^2+.0997D^3+.1199D^4-.2285D^5}$	−23.2045
ARMA(5, 5)	$\frac{1.0164-.8304D-.1269D^2-.0877D^3-.1418D^4+.2508D^5}{1-1.6394D+.4342D^2+.2137D^3-.0121D^4+.0076D^5}$	−32.456
ARMA(6, 6)	$\frac{1.0164-.7291D-.2078D^2-.0991D^3-.152D^4+.2432D^5+.0236D^6}{1-1.5397D+.2726D^2+.2566D^3+.0065D^4+.0141D^5-.0055D^6}$	−35.0359

Table 1: ARMA approximations for $w(D)$ of Example 1 and their achievable NNTEs

Example 1 (UEC)

Under the UEC, it can be checked from Figure 3 that comparable performance (an output SNR of about 26 dB) results from the number-of-tap choices ($N_b = 8, N_f = 16$) and ($N_b = 2, N_f = 120$). For the latter choice, the optimum TIR is given by $b_{opt}(D) = D^{97}(.7158 + .5722D + .4003D^2)$, i.e., $\Delta_{opt.} = 97$. The data rate reduction factor of a DMT with a block size of 128 is reduced from %5.88 with $N_b = 8$ to %1.54 with $N_b = 2$. Since the peak sample of the equalizer impulse response (IR) in this case occurs at the end, we shall reverse the IR [5] before modeling it by a pole-zero filter to achieve better filts [5].

The transfer functions of some of the best obtained pole-zero approximations of the reversed equalizer IR, denoted by $w(D)$, together with their achievable normalized norm tap error ($\text{NNTE} \stackrel{def}{=} \frac{|\mathbf{w}-\hat{\mathbf{w}}|^2}{|\mathbf{w}|^2}$) are given in Table 1. In Figure 4, we have plotted the IR of $w(D)$ and its pole-zero approximations.

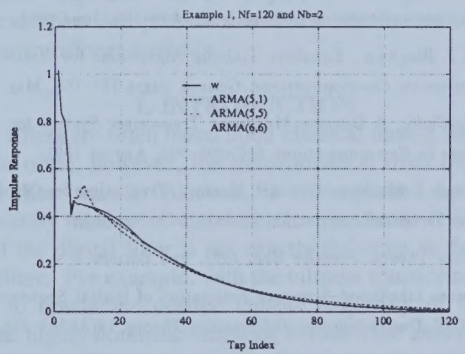


Figure 4: Pole-zero approximations for the 120-tap equalizer of Example 1

It is evident from our simulations that the generated pole-zero models accurately approximate the long FIR response of the equalizer at a great reduction in the number of filter coefficients. Therefore, our proposed pole-zero input-aided equalizer achieves the two desirable (and usually conflicting!) objectives of high performance and low implementation cost.

5. Conclusions

We derived a unified computationally-efficient method for computing the optimum finite-length coefficients of the MMSE-IAE and the resulting conditioned TIR, under unit-tap and unit-energy constraints on the TIR.

The presented approach accommodates the general cases of correlated input and noise sequences and fractionally-spaced equalizers. In addition, it simplifies the task of characterizing and optimizing the delay parameter that

[5] Reversal of the IR coefficients has the effect of replacing the zeros of the filter response by their reciprocals [15].

plays a crucial role in optimizing performance when finite–length filters are used.

Finally, we illustrated through computer simulations the viability of having a long FIR MMSE–IAE to achieve high–performance with a short target response that, in turn, enables a practical implementation of ML receivers. A stable pole–zero model of the long FIR MMSE–IAE was then computed and shown to approximate accurately the MMSE–IAE impulse response with much fewer coefficients, resulting in significant complexity reductions.

References

[1] H. Akaike. Block Toeplitz Matrix Inversion. *SIAM J. Appl. Math.*, 24(2), March 1973.

[2] N. Al-Dhahir and J. Cioffi. Fast Algorithms for the Computation of the FIR MMSE-DFE. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages IV533–IV536, San Francisco, March 1992.

[3] N. Al-Dhahir and J. Cioffi. MMSE Decision Feedback Equalizers : Finite-Length Results. *IEEE Transactions on Information Theory*, 1994. To appear.

[4] N. Al-Dhahir and J. Cioffi. Optimum Finite-Length Equalization for Multicarrier Transceivers. In *Globecom Conference Proceedings, San Francisco, CA*, November 1994.

[5] N. Al-Dhahir, A. Sayed, and J. Cioffi. A High-Performance Cost-Effective Pole-Zero MMSE-DFE. In *Allerton Conference on Communication, Control, and Computing*, pages 1166–1175, September 1993.

[6] J. Chow and J. Cioffi. A Cost-Effective Maximum Likelihood Receiver for Multicarrier Systems. In *International Conference on Communications, Chicago*, pages 948–952, June 1992.

[7] J. Chow, J. Cioffi, and J. Bingham. Equalizer Training Algorithms for Multicarrier Modulation Systems. In *International Conference on Communications, Geneva*, pages 761–765, May 1993.

[8] J. Chow, J. Tu, and J. Cioffi. A Discrete Multitone Transceiver System for HDSL Applications. *IEEE Journal on Selected Areas in Communications*, 9(6):895–908, August 1991.

[9] D. Falconer and F. Magee. Adaptive Channel Memory Truncation for Maximum Likelihood Sequence Estimation. *Bell System Technical Journal*, 52(9):1541–1562, November 1973.

[10] S. Haykin. *Adaptive Filter Theory*. Prentice Hall, 1991. 2nd Edition.

[11] G.D. Forney Jr. Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference. *IEEE Transactions on Information Theory*, 18(3):363–378, May 1972.

[12] F. Magee. A Comparison of Compromise Viterbi Algorithm and Standard Equalizatin Techniques Over Band-Limited Channels. *IEEE Transactions on Communications*, COM-23(3):361–367, March 1975.

[13] D. Messerschmitt. Design of a Finite Impulse Response for the Viterbi Algorithm and Decision-Feedback Equalizer. In *International Conference on Communications*, pages 37D.1–37D.5, Minneapolis, June 1974.

[14] M. Mueller and J. Salz. A Unified Theory of Data-Aided Equalization. *Bell System Tech. J.*, 60(9), November 1981.

[15] A. Oppenheim and R. Schafer. *Discrete-Time Signal Processing*. Prentice Hall, 1989.

[16] S. Qureshi and E. Newhall. An Adaptive Receiver for Data Transmission over Time–Dispersive Channels. *IEEE Transactions on Information Theory*, 19(4):448–457, July 1973.

[17] R. Ziegler, N. Al-Dhahir, and J. Cioffi. Nonrecursive Adaptive Decision-Feedback Equalization from Channel Estimates. In *Proceedings of IEEE Vehicular Technology Society 42nd VTS Conference*, volume 2, pages 600–603, Denver, May 1992.

DESIGN OF EXACT CLASSICAL DIGITAL FILTERS

Larry D. Paarmann and Hsiang-Chen Huang

Department of Electrical Engineering
Wichita State University
Wichita, KS 67260-0044

ABSTRACT

In this paper, a new method of designing digital filters based on an analog prototype is presented. The new method is unique in that it produces a magnitude frequency response that exactly follows that of the analog prototype, unlike that of the impulse-invariance and bilinear transform methods. This technique utilizes an algorithmic approach, involving cepstral processing to perform decorrelation of the magnitude frequency response to obtain the impulse response, followed by an algorithm to obtain the pole/zero model of appropriate order from the impulse response. In the paper, the design method is developed and described, and two examples are presented and the results are compared with impulse-invariance and bilinear transform methods.

I. INTRODUCTION

Design of digital filters are often based upon classical analog filters such as Butterworth and Chebyshev. Most often, however, the transformation from the analog prototype filter into $H(z)$, the digital filter transfer function, is accomplished by means of the impulse-invariance method, or by means of the bilinear transform. With either method, the frequency response of the digital filter is not exactly the same as the frequency response of the analog prototype filter. For example, with the bilinear transform, frequency pre-warping is required in order to obtain the desired cut-off frequency of the digital filter; this is required because of the highly nonlinear mapping between the analog frequency axis and the digital filter frequency axis. Obviously, this nonlinear mapping will distort the shape of the frequency response of the digital filter relative to the analog prototype.

One reason why it may be desirable to preserve the exact frequency response, is that it is often very simply expressible. For example, suppose a digital filter magnitude-squared frequency response could be expressed as follows:

$$|H(e^{j\omega})|^2 = \frac{1}{1 + (\omega/\omega_c)^{2n}} \quad , \tag{1}$$

where n is the integer order of the filter. Then we could say that the digital filter is an exact Butterworth design. Note that the frequency response of a digital filter based upon a Butterworth prototype, but using the impulse-invariance method, or the bilinear transform, would be a much more complicated expression than the one given above.

Another reason why it may be desirable to preserve the exact frequency response, is that well-known order equations developed for use in analog filter design would apply to the design of digital filters as well. Frequently the required order of a digital filter is determined on a trial and error basis. However, the required order for the design of an exact classical digital filter could be simply computed as a one-time routine step in the design procedure.

One reason why such filter design methods have not been developed and used before, to produce a digital filter magnitude frequency response that is exactly the same as the analog prototype, is that the paper and pencil design method appears to be intractable. However, this present paper presents an algorithmic approach, involving cepstral processing, that determines the filer design parameters.

In this paper, the cepstral processing algorithm for the filter design will be developed and explained, an example of an exact Butterworth filter design, and also of an exact Chebyshev, will be presented. Also, the relationship between the poles and zeros of impulse-invariance designed filters, bilinear transform designed filters, and exact filters designed with the cepstral processing method will be compared.

II. EXACT DESIGN PRINCIPLES

If the magnitude-squared frequency response of an analog prototype filter is cast in the form of a digital filter, as shown, for example, in (1), then the autocorrelation function of the filter impulse response could be found by means of the inverse Fourier transform of the magnitude-squared frequency response:

$$r_h(\tau) = F^{-1}\{|H(e^{j\omega})|^2\} \quad .$$

However, the impulse response, $h(n)$, corresponding to $r_h(\tau)$, is not unique. If the following assumptions are made, then $h(n)$ is unique: $h(n)$ is real, causal, and minimum phase.

Given that $|H(e^{j\omega})|^2$ is equivalent to a classical analog prototype filter, and given the above three assumptions, then the problem reduces to determining what $h(n)$ is, and given $h(n)$, what are the required parameters of $H(z)$ to correspond to $h(n)$. These two processing steps [determining $h(n)$ and then $H(z)$] are considered in the following two sections.

III. DECORRELATION OF $r_h(\tau)$

Deconvolution of signals by means of cepstral processing is well known. The decorrelation of $r_h(\tau)$ can be accomplished by similar means, and will yield $h(n)$, since

$$r_h(\tau) = E\{h(n)\,h(n+\tau)\} = \sum_{n=0}^{\infty} h(n)\,h(n+\tau) \quad .$$

Let $|H(k)|^2$ be frequency samples of $|H(e^{j\omega})|^2$. Then, $H^*(k)H(k) \Longleftrightarrow r_h(\tau)$, $H^*(k)H(k) \Longleftrightarrow \bar{h}(n) * h(n)$, $h(n) \Longleftrightarrow H(k) \Longleftrightarrow H(z)$, and $\bar{h}(n) \Longleftrightarrow H^*(k) \Longleftrightarrow H(1/z)$. Therefore, $\bar{h}(n) = h(-n)$. Since $h(n)$ is causal and minimum phase, $\bar{h}(n) = h(-n)$ is maximum phase. The problem is now seen to be one of separation of the minimum phase signal, $h(n)$, and the maximum phase signal, $\bar{h}(n)$, which requires deconvolution, since $r_h(\tau) = \bar{h}(n) * h(n)$.

Following Oppenheim and Schafer [1, Section 12.6.4], homomorphic filtering can be used for this separation. See Figure 1 for a block diagram illustration. In the figure, D_* is the characteristic system for convolution, D_*^{-1} is its inverse, $\hat{r}(n)$ is the complex cepstrum of $r_h(\tau)$, $l_{mn}(n)$ is the minimum phase lifter, $l_{mx}(n)$ is the maximum phase lifter, $h(n)$ is the minimum phase part of $r_h(\tau)$, and $h(-n)$ is the maximum phase part of $r_h(\tau)$.

The computation of $\hat{r}(n)$ is illustrated in Figure 2. Note, however, that in this application, $R(k)$ is zero-phase (real), since $r_h(\tau)$ has even symmetry. Therefore, $\hat{R}(k) = ln[R(k)]$ is a real operation. Also note that $R(k) = |H(k)|^2$, and therefore there is no need to compute the inverse Discrete Fourier transform in order to obtain $r_h(\tau)$; $R(k)$ is therefore the starting point. Note, also, that since $R(k)$ is real and even in this application, so is $\hat{R}(k)$, and therefore so is $\hat{r}(n)$. In this application of homomorphic filtering, there is, therefore, no need for any phase unwrapping.

Since $h(n)$ and $h(-n)$ overlap at one point, specifically, at $n = 0$, the lifter $l_{mn}(n)$ in Figure 1 is defined as follows:

$$l_{mn}(n) = \begin{cases} 0, & n < 0 \\ \frac{1}{2}, & n = 0. \\ 1, & n > 0 \end{cases}$$

Transforming $\hat{r}_{mn}(n)$ into $h(n)$ is illustrated in Figure 3. The processing to obtain $h(n)$ is summarized as follows:

- $R(k) = |H(k)|^2$,
- $\hat{R}(k) = ln[R(k)]$,
- $\hat{r}(n) = DFT^{-1}[\hat{R}(k)]$,
- $\hat{r}_{mn}(n) = \hat{r}(n)\, l_{mn}(n)$,
- $\hat{R}_{mn}(k) = DFT[\hat{r}_{mn}(n)]$,
- $R_{mn}(k) = exp[\hat{R}_{mn}(k)]$,
- $h(n) = DFT^{-1}[R_{mn}(k)]$.

The above processing very effectively and accurately obtains $h(n)$ from $|H(k)|^2$. The remaining step is to obtain $H(z)$ from $h(n)$.

IV. DETERMINATION OF FILTER PARAMETERS

Since $h(n)$ is causal and minimum phase, all poles and zeros will be inside the unit circle, and the number of each will be the same. Therefore, (2) follows. That is, it is known *a priori* that the number of poles and zeros is the same.

$$H(z) = \frac{\sum_{i=0}^{M} b_i z^{-i}}{1 + \sum_{i=1}^{M} a_i z^{-i}} \quad , \tag{2}$$

Note, however, that if $H(z)$ is obtained by taking the z transform of $h(n)$, the following would result: $\beta_0 + \beta_1 z^{-1} + \beta_2 z^{-2} + \cdots$, where the β values would be the unit sample

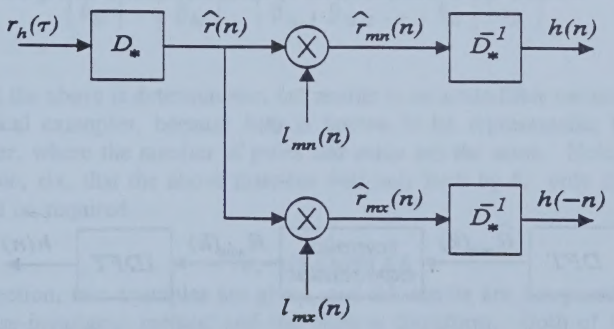


Figure 1. Homomorphic system for the separation of the minimum phase part and maximum phase part of the autocorrelation $r_h(\tau)$ [1].

response values. That is, $\beta_0 = h(0)$, $\beta_1 = h(1)$, $\cdots$. However, since $h(n)$ was obtained from an analog prototype filter, and because of the well-known properties of the transformation from the s domain to the z domain, $h(n)$ must be capable of being modeled as the response from a rational transfer function with an equal number of poles and zeros, as mentioned above, of order similar to, if not the same as, the order of the analog prototype. Therefore, the problem of determining the filter coefficients of (2), is the problem of converting the β parameters, that is the $h(n)$ values, into the a_i and b_i parameters of (2). The following procedure to accomplish the above is a modification of the work of Graupe, Krause and Moore [2]:

$$\beta_0 + \beta_1 z^{-1} + \beta_2 z^{-2} + \ldots = \frac{b_0 + b_1 z^{-1} + \ldots + b_M z^{-M}}{1 + a_1 z^{-1} + \ldots + a_M z^{-M}} \tag{3}$$

By cross multiplying and collecting terms, (3) can be simplified as follows:

$$\beta_0 + (\beta_1 + \beta_0 a_1) z^{-1} + (\beta_2 + \beta_1 a_1 + \beta_0 a_2) z^{-2} + \ldots$$

$$= b_0 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_{\cdot} z^{-M} \quad .$$

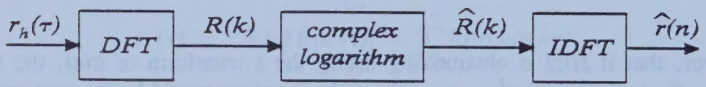


Figure 2. Details of the characteristic system, D_*, shown in Figure 1 [1].

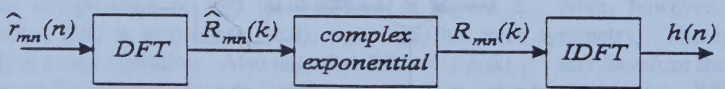


Figure 3. Details of the inverse characteristic system, D_*^{-1}, shown in Figure 1 [1].

By equating powers of z, it is clear that

$$b_0 = \beta_0$$

$$b_1 = \beta_1 + \beta_0 a_1$$

$$b_2 = \beta_2 + \beta_1 a_1 + \beta_0 a_2$$

$$\vdots$$

$$b_M = \beta_M + \beta_{M-1} a_1 + \cdots + \beta_0 a_M$$

$$0 = \beta_{M+1} + \beta_M a_1 + \cdots + \beta_1 a_M$$

$$\vdots$$

$$0 = \beta_{2M} + \beta_{2M-1} a_1 + \cdots + \beta_M a_M .$$

From the above it follows that

$$\begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ \cdot \\ a_M \end{bmatrix} = - \begin{bmatrix} \beta_M & \beta_{M-1} & \cdots & \beta_1 \\ \beta_{M+1} & \beta_M & \cdots & \beta_2 \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \beta_{2M-1} & \beta_{2M-2} & \cdots & \beta_M \end{bmatrix}^{-1} \begin{bmatrix} \beta_{M+1} \\ \beta_{M+2} \\ \cdot \\ \cdot \\ \cdot \\ \beta_{2M} \end{bmatrix} ,$$

and

$$\begin{bmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ \cdot \\ b_M \end{bmatrix} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \cdot \\ \cdot \\ \cdot \\ \beta_M \end{bmatrix} + \begin{bmatrix} \beta_0 & 0 & \cdots & 0 \\ \beta_1 & \beta_0 & \cdots & 0 \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \beta_{M-1} & \beta_{M-2} & \cdots & \beta_0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ \cdot \\ a_M \end{bmatrix} .$$

Note that the above is deterministic, but results in accurate filter parameters, as is borne out by practical examples, because $h(n)$ is known to be representable by a finite-order pole/zero filter, where the number of poles and zeros are the same. Note that if the order is, for example, six, that the above matrices will only be 6 by 6; only the first 13 values of $h(n)$ would be required.

V. EXAMPLES

In this section, two examples are given, and the results are compared to that obtained by the impulse-invariance method and the bilinear transform. Both of the examples are lowpass filters, with a 3 dB cutoff frequency of $\pi/4$.

Example 1

In Figure 4 is shown the magnitude frequency response of three 4th-order Butterworth filters. Also, for comparison, the exact 4th-order Butterworth response, according to (1), is shown. Note that the filter design according to the above cepstral processing falls directly on the exact 4th-order Butterworth response with practically no error; however, this was achieved actually with a 5th-order filter, as is discussed below.

In Figure 5 is shown pole/zero plots for the 4th-order Butterworth filters: (a) As designed by the bilinear transform method, (b) As designed by the impulse-invariance method, (c) As designed by the above cepstral processing method. Note that there are five poles and zeros in Figure 5 (c). It was observed that using only four poles and zeros exhibited some error, whereas allowing the algorithm one more pole/zero combination results in the design shown with practically no error. If one studies Figure 5, parts (b) and (c), carefully, it can be seen that the poles in the right half of part (c) are almost exactly the same as those in part (b). Part (c) shows a pole/zero pair on the real axis close to $z = -1$ that is not in part (b). It is primarily this pole/zero pair that enables the frequency response to match the ideal Butterworth response.

Example 2

In Figure 6 is shown the magnitude frequency response of three 4th-order Chebyshev filters with 3 dB of ripple. Also, for comparison, the exact 4th-order Chebyshev response is shown. Note that the filter design according to the above cepstral processing falls directly on the exact 4th-order Chebyshev response with practically no error; however, this was achieved actually with a 5th-order filter, as is noted and discussed below.

In Figure 7 is shown pole/zero plots for the 4th-order Chebyshev filters: (a) As designed by the bilinear transform method, (b) As designed by the impulse-invariance method, (c) As designed by the above cepstral processing method. Note that there are five poles and zeros in Figure 7 (c). As in Example 1, it was observed that using only four poles and zeros exhibited some small error, whereas allowing the algorithm one more pole/zero combination results in the design shown with practically no error. This is reasonable, since the pole/zero pair in the left plane almost cancel out, but the zero is slightly to the left of the pole, pulling the frequency response down slightly at higher frequencies to match the exact 4th-order Chebyshev response. If one studies Figure 7, parts (b) and (c), carefully, it can be seen that the poles in the right half of part (c) are almost exactly the same as those in part (b).

It is worth noting that when 6th-order filters are designed, similar to the above two examples, that there is no need for the exact filter design to be of higher order than six. In fact, it is noted that the exact design and the impulse-invariance design are exactly the same. That is, the 6th-order impulse-invariance design filter exactly fits the exact Butterworth response given in (1), or the exact 6th-order Chebyshev response, at least within the graphical accuracy limitations, for both the magnitude response and the position of the poles and zeros. This would also be true for orders higher than six.

VI. CONCLUSIONS

A method has been presented, whereby digital filters that have exact classical magnitude responses can be designed. Two examples have been given. Those, plus others that have been designed using the method, confirm the theory. It is interesting to note that, at least for the lowpass filters studied, the poles of the impulse-invariance design appear as poles in the exact design. It is also noted that for lowpass filters of sufficiently high order, or sufficiently low cutoff frequency, that the impulse-invariance design and the exact design become identical.

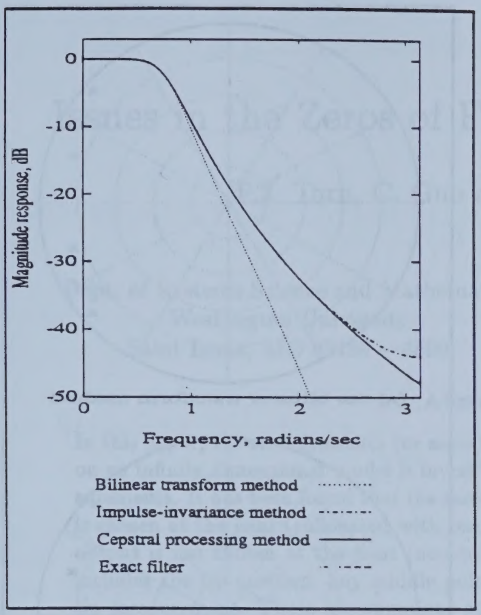


Figure 4. Magnitude frequency response for the 4th-order Butterworth filters of Example 1.

VII. REFERENCES

[1] A.V. Oppenheim and R.W. Schafer, *Discrete-Time Signal Processing*, Englewood Cliffs, NJ, Prentice-Hall, 1989.

[2] D. Graupe, D.J. Krause and J.B. Moore, "Identification of Autoregressive Moving-Average Parameters of Time Series," *IEEE Trans. Auto. Cont.*, vol. AC-20, No. 1, pp. 104-107, Feb., 1975.

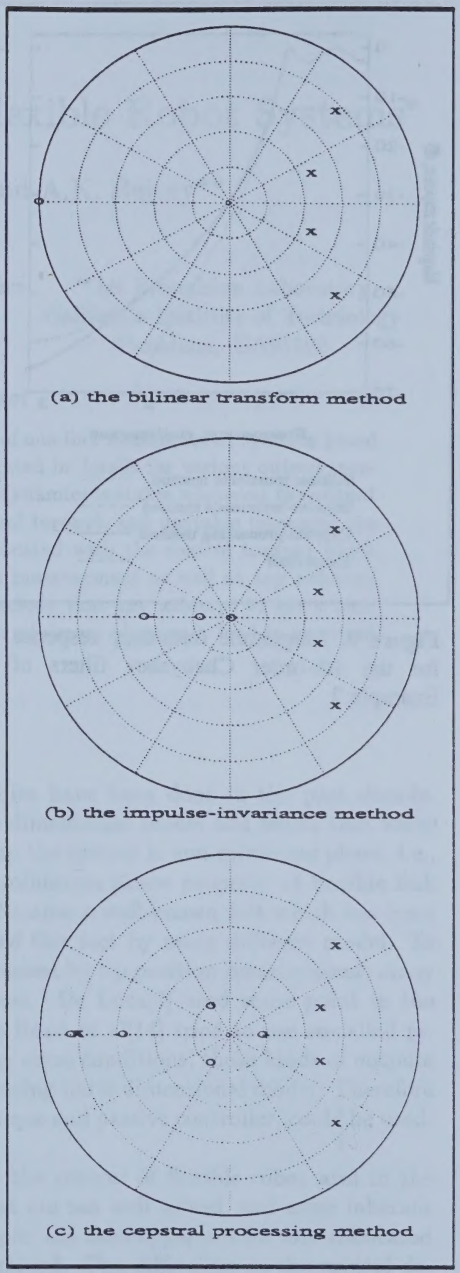


Figure 5. Pole/zero plots for the 4th-order Butterworth filters of Example 1.

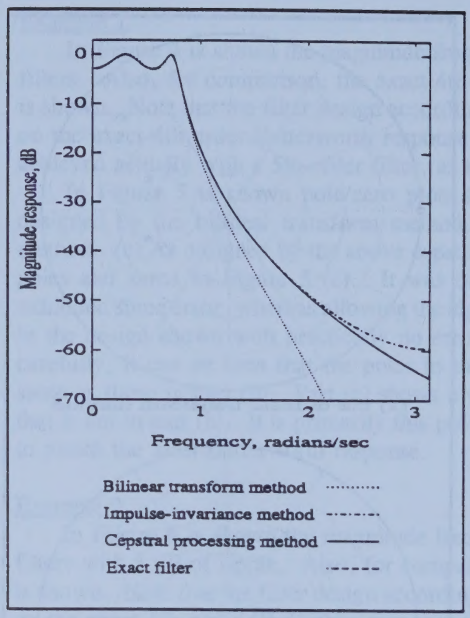


Figure 6. Magnitude frequency response for the 4th-order Chebyshev filters of Example 2.

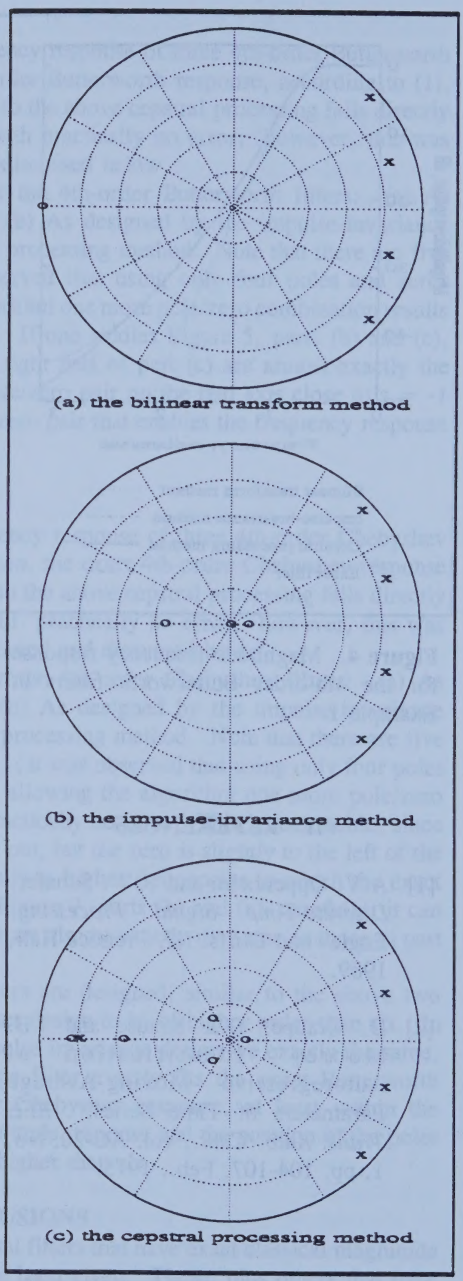


Figure 7. Pole/zero plots for the 4th-order Chebyshev filters of Example 2.

Issues in the Zeros of Flexible Robot Systems*

T.J. Tarn, C. Guo and A.K. Bejczy**

Dept. of Systems Science and Mathematics
Washington University
Saint Louis, MO 63130 – 4899

**Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA91109

Abstract

In this paper, the zero-dynamics (or zeros) of one-link flexible robot systems based on an infinite dimensional model is investigated in details for various output measurements. It has been found that the zero-dynamics is stable whenever the output is chosen at the joint (collocated with control torque), and unstable whenever the output is not chosen at the joint (non-collocated with the control torque) which includes the tip position, any middle point measurement as well as any reflected tip measurement. These investigations conclude that any attempt to use a non-collocated measurement to make the system minimum phase may lead to incorrect conclusions.

1 Introduction

For a one-link flexible robot arm, many studies have been done in the past decade. Cannon and Schmitz[1] first used linear finite dimensional model and found that when the output is chosen as the arc-length of the tip the system is non-minimum phase, i.e., the zeros are unstable. Since then, the non-minimum phase property of flexible link robot system with tip position maesurement became a well-known fact which has been cited in many articles. De Luca[3] also verified this fact by using different model. To avoid non-minimum phase property which is caused by tip position measurement, many researchers were trying to use different outputs. De Luca[3] used some point in the link other than the tip and joint angle, while Rossi et al[14] tried to use so-called reflected tip position. They all proved that under some conditions, those kinds of outputs would bring the system to minimum phase by using finite dimensional model. Therefore some control technique, such as inversion technique and passive controller, could be used.

Although there has been much work done in the control of flexible robot arm in the past decade, there are still many problems that are not well solved, and some inherant properties of flexible robot systems, for example, the zero-dynamics for non-collocated flexible link robot systems, are not well understood. The difficulties in the control design are due to the vibration of the link which has infinitely many modes and also the non-collocation of control and output measurement. The latter, of course, makes the system non-minimum phase. In this paper, the zero-dynamics of a one-link flexible robot

*This work was supported in part by the National Science Foundation under Grant IRI-9106317 and by the Sandia National Laboratories under Contract AC-3752-C.

arm will be investigated for various output measurements. To make things simple and clear, the dynamic model of the one link flexible robot considered here is a linear partial differential equation.

2 Zero Dynamics of Nonlinear Infinite Dimensional Systems

In this section, the definition of zero-dynamics for infinite dimensional nonlinear systems is given, linear infinite dimensional systems can be treated as a special case.

Consider the following distributed-parameter system:

$$\begin{aligned} \dot{q} &= f(q) + g(q)u \\ y &= h(q) \end{aligned} \tag{1}$$

where $q = (q_1, q_2, ..., q_n)'$, and $\dot{q}_i$ represents dq_i/dt if q_i depends only on t and represents $\partial q_i/\partial t$ if q_i is a function of both t and a spacial variable, say x. $f = [f_1, \ldots, f_n]'$ and $g = [g_1,\ g_2,\ \ldots\ g_n]'$ are smooth operators on the state space $\mathbf{X}, u \in \mathbf{R}$ is the input and the product of g and u is a map from $\mathbf{U} \times \mathbf{X}$ to $\mathbf{X}$, y is the output and h is a scalar function which is also a smooth operator on $\mathbf{X}$.

Definition 1 A nonlinear system has a relative degree r at q° if

$$\mathcal{L}_g \mathcal{L}_f^k h(q) = 0 \tag{2}$$

for all $k < r - 1$, and for all q in a neighborhood of q°; and

$$\mathcal{L}_g \mathcal{L}_f^{r-1} h(q) \neq 0 \tag{3}$$

where

$$\mathcal{L}_f h(q) = \frac{\partial h(q)}{\partial q} f(q),\ \mathcal{L}_f^k h(q) = \frac{\partial \mathcal{L}_f^{k-1} h(q)}{\partial q} f(q),\ \mathcal{L}_g \mathcal{L}_f^{r-1} h(q) = \frac{\partial \mathcal{L}_f^{r-1} h(q)}{\partial q} g(q).$$

The relative degree r of the system given above is with the consideration that f, g and h are operators and the Lie derivatives should be calculated using formulas given in[9]. Choose $h, \mathcal{L}_f h, \ldots, \mathcal{L}_f^{r-1} h$ as part of the new state variables, denoted by $\xi_1, \ldots, \xi_r$. Set

$$\begin{aligned} \xi_1(q) &= h(q) \\ \xi_2(q) &= \mathcal{L}_f h(q) \\ &\vdots \\ \xi_r(q) &= \mathcal{L}_f^{r-1} h(q) \end{aligned} \tag{4}$$

Apply the identity mapping to the rest of the state variable q to get η. If the mapping ϕ so obtained from q to $\begin{bmatrix} \xi \\ \eta \end{bmatrix}$ and its inverse are both sufficiently smooth around q°, then

it will be qualified as a state transformation. The system in the new coordinates $\begin{bmatrix} \xi \\ \eta \end{bmatrix}$ shall be in the following form, with the understanding that $\tilde{f}$ and $\tilde{g}$ are operators,

$$\begin{aligned} \dot{\xi}_1 &= \xi_2 \\ &\vdots \\ \dot{\xi}_{r-1} &= \xi_{r-2} \\ \dot{\xi}_r &= \mathcal{L}_f^r h + \mathcal{L}_g \mathcal{L}_f^{r-1} h \, u \\ \dot{\eta} &= \tilde{f}(\xi, \eta) + \tilde{g}(\xi, \eta) u \end{aligned} \tag{5}$$

Static state-feedback

$$u = \alpha(q) + \beta(q) v \tag{6}$$

can be employed where $\alpha(q)$ and $\beta(q)$ are given by

$$\begin{aligned} \alpha(q) &= -(\mathcal{L}_g \mathcal{L}_f^{r-1} h)^{-1} \mathcal{L}_f^r h \\ \beta(q) &= (\mathcal{L}_g \mathcal{L}_f^{r-1} h)^{-1} \end{aligned} \tag{7}$$

Then, we have

$$\dot{\xi}_r = \mathcal{L}_f^r h + (\mathcal{L}_g \mathcal{L}_f^{r-1} h) u = v$$

and the closed-loop system becomes

$$\begin{aligned} \dot{\xi} &= A\xi + Bv \\ y &= C\xi \\ \dot{\eta} &= \hat{f}(\xi, \eta) + \hat{g}(\xi, \eta) v \end{aligned} \tag{8}$$

where $\hat{f}$, $\hat{g}$ are operators,

$$A = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ & & \ddots & 1 \\ \bigcirc & & & 0 \end{bmatrix}_{(r \times r)}, \quad B = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \text{ and } C = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}.$$

Having decomposed the system into two subsystems via state-feedback, as indicated by Equation (8), one can design the reference input v to replace poles for the linear subsystem which is controllable and observable and of finite dimension. The output can be regulated in this case. However, in order for the full closed-loop system to be stable, one has to consider the second subsystem which is of infinite dimension. The stability of the full system is related to the spectrum of the zero dynamics of the system.

Definition 2 The zero dynamics of system (8) corresponding to y° is described by:

$$\dot{\eta} = \hat{f}(\xi^\circ, \eta) \tag{9}$$

where ξ° is given by

$$\xi^\circ := \xi(0) = [y^\circ \; 0 \ldots 0]'$$

Notice that the zero dynamics of a system really describes the dynamic behavior of the system when feedback u and initial condition of ξ are chosen to keep the output of the system fixed at some value y°. Zero dynamics is not an autonomous system.

3 Zero-dynamics of a one-link flexible robot

3.1 Dynamic Model of a One-link Flexible Robot

Without considering the gravity, Centrifugal force and Coriolis force, the dynamic model of a flexible robot arm with one flexible thin beam and rigid joint is given by [1][9]

$$J_h\ddot{\theta} - EIw_{xx}(0,t) = \tau, \tag{10}$$

$$\ddot{w} + x\ddot{\theta} + \frac{EI}{\rho}w_{xxxx} = 0, \tag{11}$$

with boundary conditions

$$w(0,t) = w_x(0,t) = w_{xx}(L,t) = w_{xxx}(L,t) = 0, \tag{12}$$

where $w(x,t)$ is the displacement of a point on the beam in y direction (Figure 1), $\theta(t)$ is the joint angle, $\dot{(\ )}$ denotes $\frac{\partial}{\partial t}(\ )$, $(\ )_x$ denotes $\frac{\partial}{\partial x}(\ )$, EI is the stiffness of the beam (E: Young's modulus; I: inertia), ρ is the material density of the link, τ is the motor torque, L is the length of the beam, and $J_h = J_m + \rho\int_0^L w^2(x,t)dx$ if assuming small deformation, and J_m is the motor inertia. For simplicity, we assume that EI and ρ are constants across the beam.

3.2 State Space Expression

Let $q_1 = \theta(t)$, $q_2 = w(x,t)$, $q_3 = \dot{\theta}(t)$, $q_4 = \dot{w}(x,t)$, and $u = \tau$, the above dynamic model may be shown to take in the state space expression as follows

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \begin{bmatrix} q_3 \\ q_4 \\ f_3(q) \\ f_4(q,x) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ g_3 \\ g_4 \end{bmatrix} u := f + gu \tag{13}$$

where $q = [q_1, q_2, q_3, q_4]'$. f_3, f_4, g_3 and g_4 may involve operators which are specified by

$$f_3(q) = \frac{EIq_{2,xx}(0,t)}{J_h}, \quad f_4(q,x) = -xf_3(q) - \frac{EI}{\rho}q_{2,xxxx},$$
$$g_3(q_2) = \frac{1}{J_h}, \qquad g_4(q_2,x) = -\frac{x}{J_h}. \tag{14}$$

3.3 The Output Measurements

For a one-link flexible robot arm control system, the following various kinds of output measurements (Figure 1) have been the most often used in the literatures:

(1) Joint angle :

$$y_1 = h(q) = \theta = q_1 \tag{15}$$

(2) Direct Strain [11]:

$$y_2 = h(q) = w_{xx}(0,t) \tag{16}$$

(3) Arclength of the tip :

$$y_3 = h(q) = L\theta(t) + w(L,t) = Lq_1 + q_2(L,t) \tag{17}$$

(4) Arclength of a middle point [3]:

$$y_4 = h(q) = \eta\theta(t) + w(\eta,t) = \eta q_1 + q_2(\eta,t), \quad \eta \in (0,L) \tag{18}$$

(5) Reflected tip position [12]:

$$y_5 = h(q) = L\theta(t) - w(L,t) = Lq_1 - q_2(L,t) \tag{19}$$

(6) Reflected tip position with weighted vibration [14]:

$$y_6 = h(q) = L\theta(t) - \kappa w(L,t) = Lq_1 - \kappa q_2(L,t), \kappa \in (0,1) \tag{20}$$

The first two are called collocated (output measurement) since control and output measurement are at the same place, while the last four are called noncollocated.

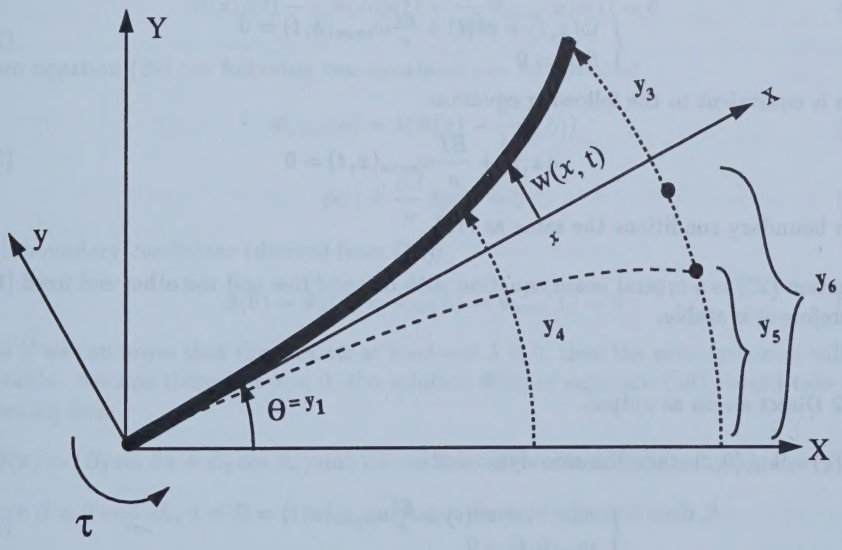


Figure 1: One-Link Flexible Robot and Its Output Measurements.

Collocated and non-collocated measurements are used for different purposes. To control the effect of vibration to the joint, collocated measurements may be used, While

to control the tip position, non-collocated measurements are much better to be used. For some other special control purposes, derivatives of about list measurements may be used, but the minimum phase property will be no change. Therefore they are not going to be listed here and neither the analysis of the zero dynamics.

For a finite dimensional linear system, it is well known that the zeros depend on the output. For a nonlinear infinite dimensional system, the zero-dynamics which plays the similar role obviously depends on the output too. Similar to the finite dimensional linear system, the system with stable zero-dynamics is called the minimum phase system, and the system with unstable zero-dynamics the nonminimum phase system.

It can be proved that the zero-dynamics of one-link flexible robot system can be obtained as following

$$\begin{cases} \ddot{w}(x,t) + x\ddot{\theta}(t) + \frac{EI}{\rho} w_{xxxx}(x,t) = 0 \\ h(\theta(t), w(x,t)) = 0 \end{cases} \tag{21}$$

with the boundary conditions the same as (12).

3.4 Zero Dynamics of Collocated Systems

3.4.1 Joint angle as output

If we choose the joint angle as the output, then the zero-dynamics is given by

$$\begin{cases} \ddot{w}(x,t) + x\ddot{\theta}(t) + \frac{EI}{\rho} w_{xxxx}(x,t) = 0 \\ \theta(t) = 0 \end{cases} \tag{22}$$

This is equivalent to the following equation

$$\ddot{w}(x,t) + \frac{EI}{\rho} w_{xxxx}(x,t) = 0 \tag{23}$$

with boundary conditions the same as (12).

Equation (23) is a typical beam equation with one end free and the other end fixed [15]. Therefore it is stable.

3.4.2 Direct strain as output

If $h(q) = w_{xx}(0,t)$, then the zero-dynamics is

$$\begin{cases} \ddot{w}(x,t) + x\ddot{\theta}(t) + \frac{EI}{\rho} w_{xxxx}(x,t) = 0 \\ w_{xx}(0,t) = 0 \end{cases} \tag{24}$$

Here $w_{xx}(0,t) = 0$ can be considered as a boundary condition. Let $v(x,t) = w(x,t) + x\theta(t)$, the zero-dynamics may be written as the following

$$\ddot{v}(x,t) + \frac{EI}{\rho} v_{xxxx}(x,t) = 0 \tag{25}$$

with boundary conditions

$$v(0,t) = v_{xx}(0,t) = v_{xx}(L,t) = v_{xxx}(L,t) = 0, \tag{26}$$

Again, equation (25) is a typical beam equation with one end free and the other end simply supported [15]. Therefore it is stable.

It can be concluded that for collocated measurements, the zero-dynamics is stable.

3.5 Zero Dynamics of Non-collocated System

3.5.1 Arc length of the tip as output

Now the output is chosen as the arc length of the tip, the zero-dynamics is

$$\begin{cases} \ddot{w}(x,t) + x\ddot{\theta}(t) + \frac{EI}{\rho} w_{xxxx}(x,t) = 0 \\ L\theta(t) + w(L,t) = 0 \end{cases} \tag{27}$$

This is equivalent to the following equation

$$\ddot{w}(x,t) - \frac{x}{L}\ddot{w}(L,t) + \frac{EI}{\rho} w_{xxxx}(x,t) = 0 \tag{28}$$

Using the separation of variables method, let $w(x,t) = \Phi(x)p(t)$ we can obtain

$$\Phi(x)\ddot{p}(t) - \frac{x}{L}\Phi(L)\ddot{p}(t) + \frac{EI}{\rho}\Phi_{xxxx}(x)p(t) = 0 \tag{29}$$

From equation (29) the following two equations can be obtained

$$\Phi_{xxxx}(x) = \lambda(\Phi(x) - \frac{x}{L}\Phi(L)) \tag{30}$$

$$\ddot{p}(t) + \frac{EI}{\rho}\lambda p(t) = 0 \tag{31}$$

with boundary conditions (derived from (12))

$$\Phi(0) = \Phi_x(0) = \Phi_{xx}(L) = \Phi_{xxx}(L) = 0 \tag{32}$$

Now if we can prove that there exists at least one $\lambda < 0$, then the zero-dynamics will be unstable. Assume there is a $\lambda < 0$, the solution $\Phi(x)$ of equation (30) should take the following form

$$\Phi(x) = (B_1 \sin\beta x + B_2 \cos\beta x)\sinh\beta x + (B_3 \sin\beta x + B_4 \cos\beta x)\cosh\beta x + B_5 x \tag{33}$$

where $\beta > 0$ and B_i, $i = 1, \ldots, 5$ are constants corresponding to each β.

Using the boundary conditions (32), the following equations can be obtained

$$\begin{cases} B_4 = 0 \\ B_2 + B_3 + B_5 = 0 \\ (-B_4 s + B_3 c)sh + (-B_2 s + B_1 c)ch = 0 \\ (-(B_2 + B_3)s + (B_1 - B_4)c)sh + (-(B_1 + B_4)s + (-B_2 + B_3)c)ch = 0 \end{cases} \tag{34}$$

where $s = \sin \beta L,\ c = \cos \beta L,\ sh = \sinh \beta L,\ ch = \cosh \beta L.$

From equations (30) and (33), one can check that

$$-4\beta^4 = \lambda \tag{35}$$

$$\Phi_{xxxx}(x) = -4\beta^4(\Phi(x) - B_5 x) \tag{36}$$

By comparing equation (36) with equation (30), the following equation can be obtained

$$B_5 = \frac{\Phi(L)}{L} = ((B_1 s + B_2 c)sh + (B_3 s + B_4 c)ch + B_5 L)/L \tag{37}$$

Equations (34) and (37) may be written in matrix form as follows

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & \beta & \beta & 0 & 1 \\ cch & -sch & csh & -ssh & 0 \\ csh - sch & -ssh - cch & -ssh + cch & -csh - sch & 0 \\ ssh & csh & sch & cch & 0 \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{38}$$

Let $V = [B_1, B_2, B_3, B_4, B_5]$, then $V \neq 0$ means that equation (38) has a nontrivial solution $\Phi(x)$ corresponding to a negative eigenvalue λ. It can be proved that $V \neq 0$ only if $\exists \beta > 0$ so that the following equation is satisfied

$$\cos(\beta L)\sinh(\beta L) + \sin(\beta L)\cosh(\beta L) = 0 \tag{39}$$

It is easy to check that there exists a $\beta > 0$ such that equation (39) is satisfied. This means that there exists a $\lambda < 0$. From equation (31), we can see that the zero-dynamics is unstable.

3.5.2 A middle point measurement

Now the output is chosen as a middle point at the link, the zero-dynamics is

$$\begin{cases} \ddot{w}(x,t) + x\ddot{\theta}(t) + \frac{EI}{\rho} w_{xxxx}(x,t) = 0 \\ \eta\theta(t) + w(\eta,t) = 0, \quad \eta \in (0,L) \end{cases} \tag{40}$$

Similar to §3.5.1, one can obtain that

$$B_5 = \frac{\Phi(\eta)}{\eta} = ((B_1 s_\eta + B_2 c_\eta)sh_\eta + (B_3 s_\eta + B_4 c_\eta)ch_\eta + B_5\eta)/\eta \tag{41}$$

where $s_\eta = \sin \beta\eta,\ c_\eta = \cos \beta\eta,\ sh_\eta = \sinh \beta\eta,\ ch_\eta = \cosh \beta\eta.$

It can be proved that $V \neq 0$ only if $\exists \beta > 0$ so that the following equation is satisfied

$$s_\eta sh_\eta(chsh - cs) - s_\eta ch_\eta ch^2 - c_\eta sh_\eta c^2 = 0 \tag{42}$$

It can be checked that there exists a $\beta > 0$ such that equation (45) is satisfied even for very small η. Therefore there exists a $\lambda < 0$. So the zero-dynamics is unstable for this case.

3.5.3 The reflected tip position

Now if the output is chosen as the reflected tip position, the zero-dynamics is

$$\begin{cases} \ddot{w}(x,t) + x\ddot{\theta}(t) + \frac{EI}{\rho} w_{xxxx}(x,t) = 0 \\ L\theta(t) - w(L,t) = 0 \end{cases} \tag{43}$$

Similarly, we can say that there exists a $\beta > 0$ which satisfies the following equation

$$sch + csh - 2L\beta(ch^2 + c^2) = 0 \tag{44}$$

so that $V \neq 0$. Therefore the zero-dynamics is unstable.

3.5.4 The weighted reflected tip position

Now if the output is the reflected tip position with weighted ratio κ, then the he zero-dynamics is

$$\begin{cases} \ddot{w}(x,t) + x\ddot{\theta}(t) + \frac{EI}{\rho} w_{xxxx}(x,t) = 0 \\ L\theta(t) - \kappa w(L,t) = 0 \end{cases} \tag{45}$$

Similarly, the following equations can be obtained

$$B_5 = -\kappa((B_1 s + B_2 c)sh + (B_3 s + B_4 c)ch + B_5 L)/L \tag{46}$$

It can be proved that $V = [B_1, B_2, ..., B_5]^T \neq 0$ only if $\exists \beta > 0$ so that the following equation is satisfied

$$sch + csh - (L + \frac{L}{\kappa})\beta(ch^2 + c^2) = 0 \tag{47}$$

Again, by using computer one can check that there exists a $\beta > 0$ such that equation (59) is satisfied. Therefore the zero-dynamics is unstable.

4 Conclusions

We concluded that when the output is chosen as a collocated measurement, the zero-dynamics is stable, while for any non-collocated output measurement, the zero-dynamics is always unstable. This conclusion contradicts the results obtained in [3] [14] [12] based on finite dimensional models. Since those finite dimensional models they used are derived from infinite dimensional model we were using except De Luca's, we can say that the infinite dimensional model are much closer to the real flexible robot system. For the control purposes, we can say that for the system with stable zero-dynamics one can use nonlinear feedback (if the system is modeled as nonlinear PDE) or passive control as well as any other control to stablize the system, while for the system with unstable zero-dynamics, the nonlinear feedback control and passive control, which utilize the minimum phase property, can not be used. If one wants to use finite dimensional model, any control method which utilizes the minimum phase property should be avoided when a non-collocated output is regulated. For multi-link flexible robot, the zero-dynamics should be similar to the one-link flexible robot case.

References

[1] Robert H. Cannon Jr., Eric Schmitz, Initial experiments on the end-point control of a flexible one-link robot, The International Journal of Robotics Research, Vol.3, No.3, Fall 1984

[2] F.Bellezza, L. Lanari, G. Ulivi, Exact modeling of the flexible slewing link, Proceedings IEEE International Conference on Robotics and Automation, Cincinnati, Ohio,pp.734-739, 1990

[3] A. De. Luca, P. Lucibello, G. Ulivi, Inversion techniques for trajectory control of flexible robot arms, Journal of Robotic Systems, 6(4),325-344(1989)

[4] David Wang, M. Vidyasagar, Transfer functions for a single flexible link, The International Journal of Robotics Research, Vol.10, No.5, Oct. 1991

[5] T. Komatsu, M. Uenohara, S. Hkura, H. Miura, I. Shimoyama, Active vibration control for flexible space environment use manipulators, Dynamics of controlled mechanical systems–edited by G. Schweitzer and M. Mansour, 1988

[6] Andrew D. Christian, Warren P. Seering, Initial Experiments with a flexible robot, Proceedings IEEE International Conference on Robotics and Automation, Cincinnati, Ohio,pp.722-727, 1990

[7] A. De Luca, Bruno Siciliano, Trajectory control of a non-linear one-link flexible arm, Int. J. Control, 1989, Vol.50, No.5, 1699-1715

[8] A. De Luca, L. Lanari, P. Lucibello, S. Panzieri, G. Ulivi, Control experiments on a two-link robot with a flexible forearm, 29th IEEEE Conference on Decision and Control, Honolulu, HI, Dec. 1990

[9] Xuru Ding, An analytic study of general dynamic model and nonlinear feedback control of flexible robot arms, Doctor's Dissertation, Dept. of Systems Science and Mathematics, Washington University, St. Louis, MO., USA, 1989

[10] Jan Bontsema, Dynamic stabilization of large flexible structures, Ph.D. Dissertation, Dept. of mathematics, University of Groningen, Neitherland, June 1989

[11] Zheng-Hua Luo, Direct Strain Feedback Control of Flexible Robot Arms: New Theoretical and Experimental Results, IEEE Transactions on Automatic Control, Vol.38, No.11, Nov. 1993, pp.1610-1622.

[12] Hemanshu Roy Pota and M. Vidyasagar, Passivity of Flexible Beam Transfer Functions with Modified Outputs, Proceedings of the 1991 IEEE International Conference on R&A, Sacramento, California, April 1991, pp.2826-2831.

[13] David Wang and M Vidyasagar, Transfer Functions for a Single Flexible Link, The International Journal of Robotics Research, Vol.10, No.5, Oct. 1991, pp.540-549.

[14] Mauro Rossi, Kai Zuo and David Wang, Issues in the Design of Passive Controllers for Flexible Link Robots, 1994 R&A

[15] S. Timoshenko, D.H. Young, W. Weaver, Jr., "Vibration Problems in Engineering," Fourth Edition, John Wiley & Sons, 1974.

Collision Avoidance via On-Line Planning and Control for Robot Manipulator in a Non-Stationary Environment

Zhenyu Yu, Bijoy K. Ghosh*, Tzyh-Jong Tarn and Chuanfan Guo
Department of Systems Science and Mathematics
Washington University, St. Louis, MO 63130

Abstract

In this paper we propose an approach to the problem of collision avoidance for a PUMA560 type of robot manipulator. We assume that the manipulator has been planned to follow a pre-specified path and that the planning phase did not incorporate the existence of certain (stationary or non-stationary) obstacles in the environment. It is proposed to alter the acceleration and velocity profile of the manipulator on-line along the existing path to guarantee safe, collision-free manipulation as and when the obstacles are detected (possibly with the aid of a vision system). Line of sight distance and maximum speed of the motion of obstacle are computed from visual information and the commanded acceleration and speed of the manipulator are altered so that if it were to move towards the obstacle along the line of sight, it can stop safely with existing bounded control before the occurence of collision. The proposed scheme has been demonstrated on a PUMA560 robot with an Intelledex vision processor and two CCD cameras.

1 Introduction

In this paper we propose an on-line algorithm that would enable a robot manipulator to follow a pre-specified trajectory safely. The obstacles are assumed to be spherical objects that can be fixed or moving with known maximum speed. The important thing is that the path of the manipulator has been originally planned without paying attention to the obstacles. As these obstacles are detected we propose to alter the motion plan in order to avoid collision.

We assume that as the manipulator moves along a specified trajectory, following a pre-planned control scheme, there is a *vision system* which constantly monitors the environment looking for obstacles. As soon as an obstacle is detected, the actual speed of robot is compared with some *safety bounds* computed on-line based on the *line of sight* distance and the maximum speed of obstacle. If the actual speed is within the safety bounds, the manipulator will continue following its original plan. Otherwise, the existing plan is altered to satisfy the safety bounds. This is done essentially to reduce the speed of manipulator along the trajectory. The safety bounds are derived by requiring the speed of the end effector be such that if it were to move with the current speed as initial speed along the line of sight towards an obstacle, it can be stopped (with the application of existing bounded controls) before occurrence of collision. This scheme would ensure that the movement of the end-effector be collision-free. The safety of entire manipulator is achieved by representing the manipulator with unions of spheres and extending the proposed planning scheme to safety of multiple moving objects.

The problem of *collision avoidance* has been studied intensively as an integral part of *robot motion planning*. The basic goal of robot motion planning is to design a trajectory for the manipulator to go from an initial configuration to a final configuration in a work-space possibly populated with obstacles [1][2][3]. Typically, planning would involve specifying the *path* of robot motion and the *velocity and acceleration profiles* along the path [17].

A major difficulty in path planning comes from the requirement that the motion of entire manipulator be safe during the operation. Typical work in this area includes the work of Lozano-Perez [1][2] on the concept of configuration-space approach, the work of Khatib [5] on potential field approach and the work of Kalaycioglu et al [4] on the approach of virtual impedance control. Excellent survey can be found in [3], [16] and [17]. Other schemes have also been developed to detect collision between manipulator and obstacles [9][10][13][14].

* Partially supported by DOE under grant No. DE-FG02-90ER14140

Collision detection is often used in a design-and-verify kind of planning strategy.

However, the problem of preventing collision from happening between robot and non-stationary obstacles during operation remains open. Most work on collision avoidance in existence of moving obstacles has assumed the obstacle motion is known [6]-[10][12]. Some kind of collision detection scheme is then developed and hope correction can be done in time to avoid the collision if it were to happen. However collision avoidance is not guaranteed.

Because of the on-line collision avoidance scheme that is to be discussed in this paper, the complete collision avoidance problem can better be solved in two steps. In the first step, a safe trajectory is designed for the manipulator to accomplish an operation according to the task definition while taking into account of possibly simplified structure of work-space. In the second step, our on-line collision avoidance scheme can then be employed to tackle things not considered before. But, completely re-design the trajectory on-line is impractical. What we can do however is to produce or to modify the velocity profile to control how fast the robot moves along the path such that the robot will never run into any obstacle.

2 Determination of Safety Bounds

In this section we establish the safety bounds for a body moving along a specified path when an obstacle exists in the environment. It takes time to stop a moving body even when maximum thrust available is exerted upon it against its motion. So, there is a minimum distance the body has to travel before it comes to a stop which is called stopping distance. If an obstacle is in the path of body motion and is at a distance less than this distance, collision between the two would be inevitable. We say the body is *safe* if it can stop approaching the obstacle before colliding into it.

2.1 Safety Bounds for a Moving Body

Assume that the path of motion of a controlled bodyhas been pre-specified which can be represented by $X(s)$, where s is the *curve length* along the path. Then the motion can be described by

$$\dot{X} = X_v = T(s)v \qquad \dot{X}_v = X_a = \Gamma(s)v^2 + T(s)a \tag{1}$$

where $T(s) = X'(s)$ and $\Gamma(s) = X''(s)$ are, respectively, the tangent vector and the curvature of the path at s, while v and a are, respectively, the speed and acceleration of the body along the path satisfying

$$\dot{s} = v \qquad \dot{v} = a \tag{2}$$

Since $T(s)$ and $\Gamma(s)$ both depend only on $X(s)$, the motion of the body along $X(s)$ is completely described by Eq. (2) which can be thought of as a dynamical system with input a. Since the thrust is assumed to be limited we have $|a| \le a_m$, where a_m is the maximum acceleration corresponding to maximum thrust. Similarly, v has to satisfy constraint $|v| \le v_m$ for other safety reasons we will see later. So the motion planning problem along a specific path is reduced to determining input a or, equivalently, v for each s.

Assume there is a static obstacle in the environment (Fig. 1). which may or may not be in the path of motion of the body. When it's in the path, collision between the two is obviously possible. However for an obstacle not in (but is close to) the path, collision is also possible. This is because, first, the position of the obstacle can be undetermined and, secondly, it's possible that the path is not followed strictly, causing potential collision between the body and

obstacles close to the path (Fig. 1).

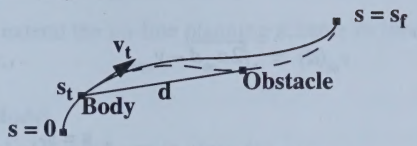


Fig. 1. Obstacle not in the Path

Assume v_t is the current speed of the moving body, t_m, the minimum time for it to come to a stop and Δs_m, the stopping distance. Based on Eq. (2), it's easy to get $\Delta s_m = \frac{1}{2a_m} v_t^2$. Therefore, given a Δs, a maximum safe speed v_m the moving body can have is given by

$$v_m(\Delta s) = \sqrt{2a_m \Delta s} \tag{3}$$

We call this relationship (Fig. 2) the *safety bounds* between v_t and Δs.

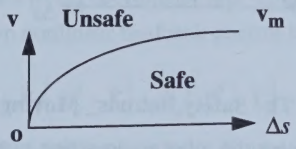


Fig. 2. The Safety Bounds

If v is bigger than $v_m(\Delta s)$, collision will happen inevitably. And we are guaranteed that collision will never happen if we can maintain $v_t \le v_m(\Delta s)$ all the time because of the fact $v_m(\Delta s) = 0$ which means the body will automatically come to a stop by the time it reaches the obstacle.

However, the path distance needed to compute the safety bounds is not available at the current time since it depends on things that will happen in the future. But, we can use normal Euclidean distance d between the body and obstacle in instead of path distance Δs to calculate the bounds. Thus Eq. (3) becomes

$$v_m(d) = \sqrt{2a_m d} \tag{4}$$

which is more conservative since Euclidean distance is always less than or equal to path distance.

Now, consider a moving obstacle with speed v_{ob} (Fig. 3). The worst thing that could happen is the obstacle being in the actual path of motion of the moving body, the direction of obstacle motion being opposite to that of the body and the magnitude of velocity of the obstacle being its maximum speed v_{om}, in which case, the speed of approaching between the two will be given by $v_a = v_t + v_{om}$. So, we have $\Delta s_m = \frac{1}{2a_m}(v_t + v_{om})^2$ which is now the distance reduced by the time the approaching speed gets to zero.

Given a path distance Δs between the body and the obstacle, the maximum speed for the body is then given by $v_m(\Delta s) = \sqrt{2a_m \Delta s} - v_{om}$. Replacing Δs by d gives us which is Eq. (4)

offset by v_{om} (Fig. 4).

$$v_m(d) = \sqrt{2a_m d} - v_{om} \tag{5}$$

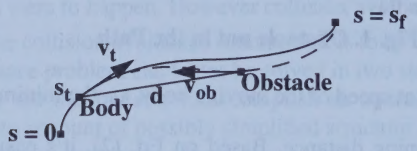


Fig. 3. Obstacle not in the Path

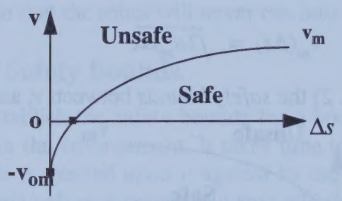


Fig. 4. The Safety Bounds_ Moving Obstacle

Because, first, we don't expect the robot to do active maneuvering to avoid the obstacle and, secondly, the *envent-based* planning and control scheme used in the original plan won't allow the manipulator to go backward [18], we assume that $v_{om} \leq \sqrt{2a_m d}$.

2.2 Safe Motion Plan for a Controlled Body

We like the body to move as fast as possible to reduce the cycle time. Thus, we will command the body to move with maximum speed allowed by the safety bounds. Let n be the number of obstacles in the environment and $v_{m\,i}$ be the maximum speed allowed by safety bounds computed for i^{th} obstacle at the moment. Then, the desired speed for the body is given by $v = min\{v_{m\,i}\}$ and the implied acceleration command is given by $a = \dot{v}$.

Assume j^{th} obstacle is the one that gives rise to the smallest maximum allowable speed. Then the velocity and acceleration commands are, respectively given by $v = \sqrt{2a_m d_j} - v_{omj}$ and $a = a_m \dfrac{v\cos(\alpha_j) + v_{oj}\cos(\beta_j)}{v + v_{omj}}$, where α_j is the angle between the velocity of the body and the sight line that goes from the body to the obstacle and β_j is the angle between the direction of motion of the body and the sight line.

When α_j is less than zero and is small in magnitude, which means the body is heading toward the obstacle, the body will reduce its speed of motion with maximum deceleration. On the other hand, if α_j is bigger than zero and is small in magnitude, which means the body is running away from the obstacle, the body will increase its speed of motion with maximum acceleration. But, the magnitude of commanded acceleration is not going to exceed the maximum acceleration allowed.

3 On-Line Motion Planning for Safety of Robot

In this section, we extend the on-line planning scheme to robot following of a predefined path for its end-effector.

3.1 Robot Dynamic Model

The dynamic model of a robot arm is given by

$$D(q)\ddot{q} + C(q, \dot{q}) + G(q) = \tau \tag{6}$$

where q, $D(q)$, $C(q, \dot{q})$, $G(q)$ and τ are, separately, the joint angle vector, the inertia matrix, the load related to centripetal and Coriolis forces, the load related to gravity, and the joint torque vector. The joint torque has to satisfy the following constraints

$$\tau_{imin}(q, \dot{q}) \le \tau_i \le \tau_{imax}(q, \dot{q}) \qquad i = 1, 2, \ldots 6 \tag{7}$$

The output is given by $Y = H(q) = (X, \theta)^t$ where $X = (x, y, z)^t$ represents the position of robot end-effector in Cartesian world coordinates and θ, the orientation.

We consider the well known nonlinear feedback control law [19] is given by

$$\tau = DJ^{-1}(q)\left(Y_a^d + K_v\left(Y_a^d - \dot{Y} \right) + K_p\left(Y^d - Y \right) - \dot{J}(q)\dot{q} \right) + C + G \tag{8}$$

where $J(q)$ is the Jacobian of $H(q)$ with respect to q, K_v and K_p are the velocity and position feedback gains, and Y_a^d, Y_v^d and Y^d are the desired linear and angular acceleration, velocity and position respectively. It can be shown that if Y_a^d, Y_v^d and Y^d satisfy the constraints $\dot{Y}^d = Y_v^d$ and $\dot{Y}_v^d = Y_a^d$, then, by proper choice of K_v and K_p, the closed-loop system can be made stable and the tracking error will vanish asymptotically.

3.2 Motion Planning for Robot End-effector

The problem of motion planning is to design Y_a^d, Y_v^d and Y^d to carry out a specific task. For PUMA 560 the orientation of end-effector can be achieved by operating only the last three joints that are very compact in size. We will neglect the orientation of end-effector and treat the last three joints together with the end-effector as one spherical body. Thus, we will only consider designing X_a^d, X_v^d and X^d.

Since we have assumed that a path has been given, our problem is reduced to one similar to what is stated in Sec. 2.1. Therefore we need first establish limit on input a and state v based on $X(s)$, the robot dynamic model given by Eq. (6) and the torque limits given by Eq. (7).

3.3 Limit on Magnitude of End-effector Velocity and Tangential Acceleration

Since $X(s)$ has been specified, q, $\dot{q}$ and $\ddot{q}$ can all be expressed as functions of s. Substituting them into Eq. (6) gives us robot dynamics along the path [15]

$$F(s)\ddot{s} + E(s, \dot{s}) = \tau \tag{9}$$

Or, equivalently, $f_i(s)\ddot{s} + e_i(s, \dot{s}) = \tau_i \qquad i = 1, 2, \ldots 6.$

3.3.1 The Situation $f_i(s) \neq 0$

Based on the inequality in Eq. (7), when $f_i(s) \neq 0$, $\ddot{s}$ must satisfy $l_i(s, \dot{s}) \leq \ddot{s} \leq u_i(s, \dot{s})$, where $l_i(s, \dot{s})$ and $u_i(s, \dot{s})$ are $(\tau_{imax} - e_i)/f_i$ or $(\tau_{imin} - e_i)/f_i$ depending on the sign of f_i. Therefore, $\ddot{s}$ must lie in the intersection of all $[l_i(s, \dot{s}), u_i(s, \dot{s})]$. If we let $l(s, \dot{s}) = \max\{l_i\}$ and $u(s, \dot{s}) = \min\{u_i\}$, where the maximum and minimum are taken over those i for which $f_i(s) \neq 0$. Then, for an admissible $\ddot{s}$ to exist, we must have

$$l(s, \dot{s}) \leq u(s, \dot{s}) \tag{10}$$

However, it may happen that, for some s and $\dot{s}$, these intervals do not overlap, *i.e.*, $l(s, \dot{s}) > u(s, \dot{s})$, which would mean the torque limit constraints of at least one joint motor would have to be violated no matter what value $\ddot{s}$ would take. Thus the manipulator would not still be able to hold the end-effector on the path. By solving the inequality in Eq. (10), we get the range of $\dot{s}$, $[v_{min}(s), v_{max}(s)]$, based on the torque limit of those motors for which $f_i(s) \neq 0$. While the admissible $\ddot{s}$, if exists, will be given by

$$l(s, \dot{s}) \leq \ddot{s} \leq u(s, \dot{s}) \tag{11}$$

3.3.2 When $f_i(s) = 0$

When $f_i(s) = 0$, the tangential acceleration of end-effector does not contribute to the load of i^{th} joint motor. However we still have to make sure that $e_i(s, \dot{s})$ be within the range allowed by Eq. (7), i.e., we must have

$$\tau_{imin} \leq e_i(s, \dot{s}) \leq \tau_{imax} \tag{12}$$

Based on this constraint, we can determine an interval $[\tilde{v}_{imin}(s), \tilde{v}_{imax}(s)]$ for $\dot{s}$. If $f_i(s) = 0$ holds for more than one joint motors, the intersection of all the intervals such established gives an interval $[\tilde{v}_{min}(s), \tilde{v}_{max}(s)]$ for $\dot{s}$ not to violate any of the torque limit constraints on those moors for which $f_i(s) = 0$.

3.4 Robot Motion Planning

Based on above discussion, we propose the following two algorithms to planning and control of a robot manipulator along a pre-specified path.

3.4.1 Method Based on Constant Speed and Acceleration Limit

Based on Eq. (10) and Eq. (12) and discussion in Sec. 3.3.1 and Sec. 3.3.2, an overall constant speed limit v_m and an overall constant acceleration limit a_m can be chosen for robot end-effector. The model that describes the path following motion of robot end-effector is then

given by $\dot{s}_0 = v_0$ and $\dot{v}_0 = a_0$ with constraints $|v_0| \le v_{m0} = v_m$ and $|a_0| \le a_{m0} = a_m$.

Assume that the manipulator body can be decomposed into spheres. Let end-effector be the 0^{th} manipulator sphere. Let X_i, $i = 1, 2,...m$, where $(m+1)$ is the total number of manipulator spheres, represent position of i^{th} manipulator sphere. Its path $X_i(s_i)$ where s_i is the curve length along the path of i^{th} sphere, can be obtained based on a fixed transformation between end-effector and i^{th} sphere which is given by $X_i(s_i) = P_i(X(s))$. Therefore, we have $\dot{X}_i(s_i) = T_i(s_i)\dot{s}_i = \frac{dP_i}{dX}T(s)\dot{s}$, where $T_i(s_i)$ is tangent of $X_i(s_i)$ at s_i, and

$$\begin{cases} \dot{s}_i = J_i(s)\dot{s} \\ \ddot{s}_i = J_i(s)\ddot{s} + \dot{J}_i(s)\dot{s} \end{cases} \tag{13}$$

where $J_i(s) = T_i^{-1}(s_i)\frac{dP_i}{dX}T(s)$.

Based on Eq. (13) and limits on $\dot{s}$ and $\ddot{s}$, one can determine a constant limit v_{mi} for $\dot{s}_i$ and a constant limit a_{mi} for $\ddot{s}_i$. When $J_i(s) = 0$, i^{th} manipulator sphere does not move. Therefore it will not run into any obstacle at the moment.

Assume we have a vision system that can on-line generate a representation of all obstacles in the world by union of spheres. Let the beginning and ending point of the path of end-effector be the 0^{th} and n^{th} obstacle spheres where $(n - 1)$ is the total number of obstacle spheres other than the beginning and ending points of the path. Let Z_j, $j = 1, 2,...n - 1$, be position of j^{th} obstacle sphere. Then the distance between i^{th} manipulator sphere and j^{th} obstacle sphere is given by $d_{ij} = \|X_i - Z_j\| - r_{mi} - r_{oj}$, where r_{mi} and r_{oj} are separately the radius of i^{th} manipulator sphere and the radius of j^{th} obstacle sphere. Let $v_{ij} = \sqrt{2a_{mi}d_{ij}} - v_{mj}$, which is the safety limit on speed of i^{th} manipulator sphere introduced by j^{th} obstacle sphere, where v_{mj} is the maximum speed of j^{th} obstacle sphere estimated through vision measurement. Then the desired velocity for end-effector is given by

$$v^d = \min_{i, J_i \neq 0} \{\frac{1}{J_i}\left(\min_j \{v_{ij}\}\right)\} \tag{14}$$

The desired acceleration command is given by differentiating v^d with respect to time. This is done numerically and the vector command input is given by Eq. (1) in on-line computation.

3.4.2 Sub-optimal Bang-Bang Control

In this section we propose a numerical method to determine if an obstacle has come within the stopping distance of manipulator in motion based on numerical integration. Similar approach has been proposed in [15] to solve the problem of time-optimal following of a specified path. The procedure is as follows:

1. Starting from current time, integrate forward Eq. (6) with τ being determined by Eq. (8) using minimum $\ddot{s}$ allowed by Eq. (11) obtained in Sec. 3.3.1 as desired tangential acceleration until the robot comes to a stop. End the entire operation if the end-effector has reached its destination.
2. Check if at any s during the above integration we have $\dot{s} = v_{max}(s)$, where $v_{max}(s)$ is determined in Sec. 3.3.1 for those i for which $f_i(s) \neq 0$.
3. Check if at any s during the above forward integration we have $\dot{s} = \tilde{v}_{max}(s)$ where $\tilde{v}_{max}(s)$ is determined in Sec. 3.3.2 for those i for which $f_i(s) = 0$.
4. Check if the manipulator has run into an obstacle in the environment by the time it comes to a stop.
5. If outcome of any of 2, 3 and 4 is true, command the manipulator to move with minimum acceleration; Otherwise, command the manipulator to move with maximum acceleration determined in Sec. 3.3.1.
6. Go to Step 1.

Note that in [15] Step 3 is ignored which is a mistake. Note also that this procedure is time-optimal since the manipulator is commanded to always move with maximum admissible speed.

4 Experimental Results

The above strategy has been implemented on a PUMA 560 robot at the Center for Robotics and Automation at Washington University. An Intelledex vision processor and two CCD cameras constitute the stereo vision system needed to generate measure of obstacle position. An SGI computer is used as the programming and control center for the robot control and planning. Vision system is connected to the SGI computer by self-designed interface.

An on-line planning algorithm using the first approach proposed is implemented in SGI computer along with the original event-base planning and control scheme. The robot is supposed to follow a pre-specified straight line path. An obstacle is put at different places in the environment for different experiments. The plots of experimental data are shown at the end of the paper where d represent distance between the manipulator and the obstacle, v_s, the safety bounds, and v, the actual speed of robot end-effector.

Fig. 5 shows the case when the obstacle is in the path of robot motion where it can be seen that the robot has to stop by the time the distance reaches zero. Fig. 6 shows the case when the obstacle is off but is close to the path where it can be seen that the manipulator has to slow down in precaution when it gets close to the obstacle and accelerates again after it has passed the obstacle. Fig. 7 shows the case of a moving obstacle moving in and out of the path during the experiment where it can be seen that the safety bounds change as the distance changes causing the robot to repeatedly slow down and accelerate.

Conclusion

The concept of safety bounds is established for a controlled body following a specified path. A safe motion plan for the moving body is formed via on-line planning based on safety bounds. The safety concepts established here applies to collision avoidance between manipulator and obstacles as well as collision avoidance between manipulator and the target object to be approached. Representation of manipulator and obstacles by unions of spherical objects is used to represent the manipulator and obstacles. The idea of safety bounds can then be applied to solve the problem of safety for a manipulator following a specified path.

References

[1] T. Lozano-Perez, "Automatic planning of manipulator transfer movements," IEEE Trans. on Syst. Man and Cybernetics, vol. SMC-11, no. 10, pp681-698, 1981.

[2] ------, "Spatial planning: A configuration space approach," IEEE Trans. on Computer, vol. C-32, no. 2, pp. 108-120, 1983.

[3] J.-C. Latombe, "Robot Motion Planning," Kluwer Academic Publishers, 1993.

[4] S. Kalaycioglu, M. Tandirci and D. Necsulescu, "Real-Time Collision Avoidance of Robot Manipulators for Unstructured Environments," in Proc. of IEEE Int Con on R &A, Atlanta, GA, Vol. 1, 1993, pp. 44-51.

[5] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," in The Int. Journal of Robotics Research, Vol. 5, No. 1, Spring, 1986, pp. 90-98.

[6] J. S. Yu and P. C. Muller, "An Approach to On-line Collision-Free Path Planning of Industrial Robots,"

[7] J. Reif and M. Sharir, "Motion Planning in the Presence of Moving Obstacles," Proc. 26th Annual IEEE Symposium on Foundations of Computer Science, 1985, pp. 144-154.

[8] N. Aggarwal and K. Fujimura, "Motion Planning Amidst Planner Moving Obstacles," Proc. 1994 IEEE Int. Conf. on R & A, San Diego, CA, May 8-13, 1994, pp. 2153-2158.

[9] C. A. Shaffer and G. M. Herb, "A Real-Time Robot Arm Collision Avoidance System," in IEEE Trans. on Robotics and Automation, Vol. 8, No. 2, April, 1992, pp. 149-160.

[10] Vincent Hayward, "Fast Collision Detection Scheme by Recursive Decomposition of A Manipulator Work-space," Proc. 1986 IEEE Int. Conf. on R &A, April, 1986, pp. 1044-1049.

[11] N. Ahuja, R. T. Chien, R. Yen and N. Bridwell, "Interference Detection and Collision Avoidance among Three Dimensional Objects," Pro. First Annual National Conf. on Artificial Intelligence, Stanford University, August 1980, pp. 44-48.

[12] Ming. C. Lin, Dinesh Manocha and John Canny, "Fast Contact Determination in Dynamic Environments," Proc. 1994 IEEE Int. Conf. on R & A, May, 1994, pp. 602-608.

[13] S. A. Cameron, "A Study of the Clash Detection Problem in Robotics," in Proc. of IEEE Int. Conf. on Robotics and Automation, St. Louis, MO, March 25-28, 1985, pp. 488-493.

[14] A. Foisy, V. Hayward and S. Aubry, "The Use of Awareness in Collision Prediction," in Proc. of IEEE Int. Conf. on Robotics and Automation, Cincinnati, OH, 1990, pp. 338-343.

[15] J. E. Bobrow, S. Dubowsky and J. S. Gibson, "Time-Optimal Control of Robotic Manipulators Along Specified Paths," The Int. Journal of Robotics Research, Vol. 4, No. 3, 1985.

[16] Z. Shiller, S. Dubowsky, "On Computing the Global Time-Optimal Motions of Robotics Manipulators in the Presence of Obstacles," IEEE Int. J. of R&A, Vol. 7, No. 6, Dec. 1991.

[17] Y. K. Hwang and N. Ahuja, "Gross Motion Planning_ A Survey," ACM Computing Surveys, Vol. 24, No. 3, Sept., 1992, pp. 219-292.

[18] T. Tarn, A. K. Bejczy and N. Xi, "Motion Planning in Phase Space for Intelligent Robot Arm Control," Proc. of IROS'92, Raleigh, NC, 1992, Vol. 3, pp. 1507-1514.

[19] T. Tarn, A. Bejczy, G. Marth and A. Ramadorai, "Performance Comparison of Four Manipulator Servo Schemes," IEEE Contr. Sys. Mag., Vol. 13, No. 1, Feb. 1993, pp. 22-29.

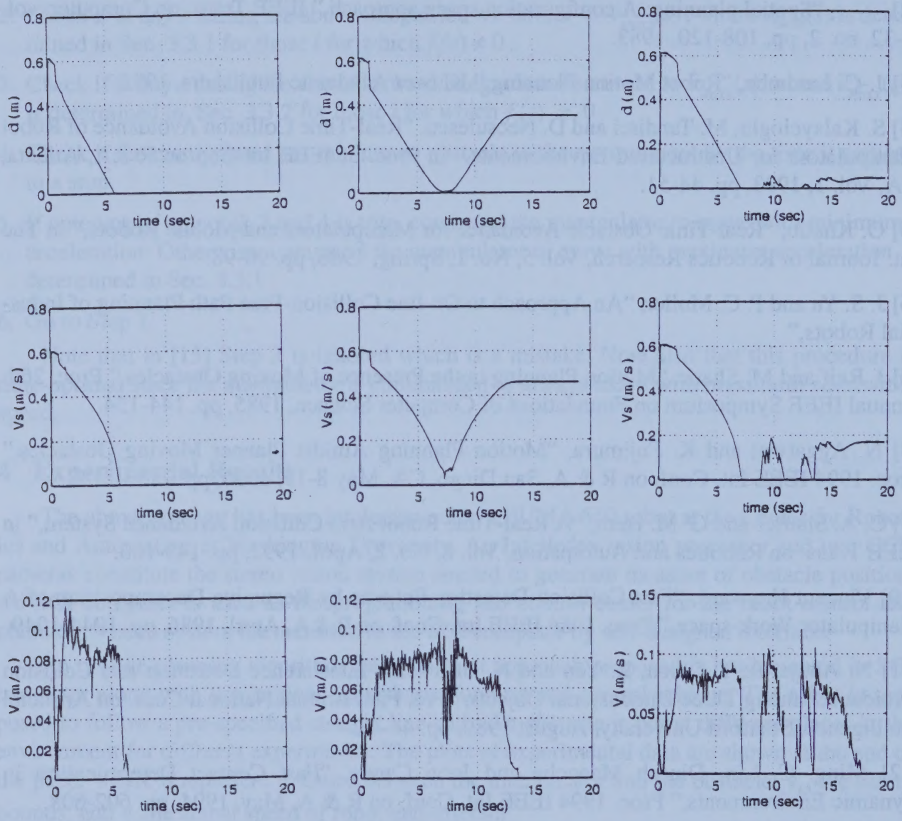


Fig. 5 Obstacle in the Path Fig. 6 Obstacle Close to the Path Fig. 7 Moving Obstacle

Action Planning and Control for Robot Phase Transition : Singular System Approach

T.J. Tarn, Y.Wu, and N. Xi

Department of Systems Science and Mathematics
Washington University
St. Louis, MO 63130 *

Abstract

Based on the singular system theory, a novel action planning and control method for robot phase transition is proposed. After introducing a sensor-referenced action planning and combining it with the nonlinear feedback linearization technique, the robot dynamic model is linearized and decoupled for the free motion mode, the phase transition mode as well as the constrained motion mode. The structure of the controller is unified for the three different action modes. More importantly, it will be able to regulate the output force after contact is established. In addition to simulation results, the scheme was implemented and tested on a 6 DOF PUMA 560 robot arm. The experimental results clearly demonstrate the advantages of the proposed method.

1 Introduction

Almost all of applications of robotic technology involve the interaction between robots and the environment. Examples can be found in assembly tasks in automated manufacturing, material handling in hazardous environment and service tasks in space. An important issue in robot action planning and control is to design a controller to achieve stable phase transition and output force regulation. The task of phase transition, sometimes called impact control can be divided into the following three operation modes: the free motion mode (or approaching mode), the phase transition mode (or impact mode), and the contact motion mode (or constrained motion mode). In the first mode, the manipulator moves from an initial point in free space following a specified trajectory heading to a transition point in the constrained surface. The constrained surface could be any kind of environment on which interaction forces/torques are exerted by the robot end effector. In the second mode, the manipulator reaches the transition point with the approaching velocity where large impact forces and possible bouncing may occur. In the third mode, after establishing contact with the environment, the manipulator must

*This research was supported in part by NSF Grant IRI-9106317, CAD-9404949 and by Sandia National Laboratories Contract No. AC 3752-C.

move along the desired path in the constrained surface with the commanded interaction forces/torques.

Control of phase transition has been a research subject for several years and various schemes have been proposed. Basically, the existing schemes can be divided into two categories. The first category is impedance control [1], [2], [3] and [4]. The impedance controller provides a unified controller structure for the three different modes. It is easy to implement. The only problem with the impedance control is that, after contact has been established, the output force cannot be regulated unless the exact environmental model was known and was integrated into the motion plan. In contrast to the impedance controller, several discontinuous control schemes have been proposed [5], [6], [7], [8], [9] and [17]. The major advantage of the discontinuous control scheme is that the output force can be regulated in the constrained motion mode. The discontinuous control schemes are designed based on the nonlinear feedback linearization technique. The linearization and decoupling are only valid when the acceleration and velocity in the commanded force direction are zero. For a constrained motion on a considerably rigid surface, this assumption is reasonable. However, in the phase transition mode, the acceleration and velocity in the constrained direction will not be zero. As a result, the linearization and decoupling cannot be achieved. Therefore, these control scheme can not be adequately applied to the phase transition mode. Furthermore, all the existing phase transition control schemes require some knowledge of the environment. Sometimes this requirement is not realistic, e.g. tasks such as dynamic grasping and material sorting. In addition, the difficulty of the phase transition control is largely due to the existence of impact between a robot and the environment. Modelling and analysis of impact phenomenon have been a research subject in mechanics. The modelling of impact between a robot and the environment has also been addressed in [10], [11] and [12].

The goal of this paper is to develop a new planning and control method for phase transition based on the Singular System Theory. It will combine the advantages of both the above methods and while overcoming their disadvantages. The dynamic model will achieve the linearization and decoupling via nonlinear feedback without the assumption that the acceleration and velocity are zero in the commanded force direction after impact. Furthermore, based on real-time force sensor information, knowledge of an environmental model is no longer required. More importantly, the structure of the controller is unified in free motion mode, phase transition mode and constrained motion mode. This provides the flexibility, efficiency and convenience to implement the scheme in a practical robotic system.

2 Feedback Linearization and Regulation for Robot Dynamic Model

The dynamic equation for a manipulator with n degree of freedom is given by

$$\begin{cases} D(q)\ddot{q} + C(q,\dot{q}) + G(q) + J(q)^T F_h = \tau \\ y = h(q) \end{cases} \tag{1}$$

where $D(q), C(q,\dot{q}), G(q)$ represent inertia, centripetal force, Coriolis force, and gravity in the joint space; $q \in \mathbb{R}^n$ is the vector of joint angles; $\dot{q}$, $\ddot{q}$ are joint space velocity and acceleration; $J(q)$ is the $n \times n$ Jacobian matrix relating joint space velocity to task space velocity, i.e., $\dot{y} = J(q)\dot{q}$; τ is the $n \times 1$ vector of joint driving torques; y is the position and orientation output in the task space; F_h is the output force in task space.

The manipulator can also be represented by task space variable as follows:

$$\Lambda(y)\ddot{y} + \mu(y, \dot{y}) + F_h = \Gamma \tag{2}$$

where $\Lambda(y), \mu(y, \dot{y})$ correspond to the inertia, centripetal force, the Coriolis force and gravity in the task space; Γ is the driving forces in the task space. This formulation could be easily derived by using the Jacobian matrix $J(q)$.

The relationships between $\dot{y}$, $\ddot{y}$, Λ, μ, Γ, and $\dot{q}$, $\ddot{q}$, D, C, G, τ are as follows:

$$\begin{aligned}
&\dot{y} = J(q)\dot{q} \\
&\ddot{y} = J(q)\ddot{q} + \dot{J}(q)\dot{q} \\
&\Lambda = J(q)^{-T} D J(q)^{-1} \\
&\mu = -J(q)^{-T} D J(q)^{-1} \dot{J}\dot{q} + J(q)^{-T} C(q, \dot{q}) + J(q)^{-T} G(q) \\
&\Gamma = J(q)^{-T} \tau
\end{aligned}$$

A state space representation can be defined as $x_1 = y$, $x_2 = \dot{y}$

$$\dot{X} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ \Lambda(x_1)^{-1} \left(\Gamma - \mu(x_1, x_2) - F_h\right) \end{bmatrix}$$

where $X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ is the $2n \times 1$ vector of state variables.

At this point, the nonlinear feedback control law [13] could be introduced as:

$$\begin{cases} \tau = DJ(q)^{-1}(V - \dot{J}(q)\dot{q}) + C(q, \dot{q}) + G(q) + J(q)^T F_h \\ \Gamma = J(q)^{-T}\tau \end{cases} \tag{3}$$

As a result, the dynamics of the robot system can be linearized and decoupled as:

$$\ddot{y} = V \tag{4}$$

where V is the commanded acceleration. In state space, we have:

$$\dot{X} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ V \end{bmatrix} \tag{5}$$

Let y_u, y_c represent the output position and orientation in unconstrained directions ("u") and constrained directions ("c"), respectively, i.e. $y = [y_u,\ y_c]^T$.

Let V_u, V_c represent the auxiliary input vector (commanded acceleration) correspondingly, i.e. $V = [V_u,\ V_c]^T$, then

$$\begin{cases} \ddot{y}_u = V_u \\ \ddot{y}_c = V_c \end{cases} \tag{6}$$

and

$$\dot{X}_u = \begin{bmatrix} \dot{x}_{u_1} \\ \dot{x}_{u_2} \end{bmatrix} = \begin{bmatrix} x_{u_2} \\ V_u \end{bmatrix} \tag{7}$$

$$\dot{X}_c = \begin{bmatrix} \dot{x}_{c_1} \\ \dot{x}_{c_2} \end{bmatrix} = \begin{bmatrix} x_{c_2} \\ V_c \end{bmatrix} \tag{8}$$

where $x_{u_1} = y_u$, $x_{u_2} = \dot{y}_u$, $x_{c_1} = y_c$, $x_{c_2} = \dot{y}_c$.

Therefore, the motion of the manipulator can be controlled independently for each direction. Obviously, the control in the unconstrained directions will not have an effect on the constrained directions and vice versa.

In the free motion mode, or in the unconstrained directions of phase transition and constrained motion modes, only position control is needed, which can be easily implemented by a PD controller [9]. However, in the phase transition and contact modes, the desired force trajectory needs be tracked in the constrained directions. More importantly, in order to avoid large impulsive forces after impact, the transient force response should also be controlled. To achieve this goal, a complete decoupling of the force and position control in the constrained directions is required. It should be noted that the decoupling in the constrained directions in the existing results [5] [14], etc. is based on the assumption that the acceleration and velocity in the constrained directions are zero. Obviously, this assumption will not hold in the phase transition mode. In other words, the force and position control in the constrained directions during the phase transition period cannot be decoupled by the above existing linearization and decoupling scheme.

Hence in order to achieve a complete force and position control decoupling, V_c has to be designed in such a way that the closed loop system (8) only contains the dynamics of force. This can be done by choosing $V_c = \dot{x}_{c_2} + u_c$ where $\dot{x}_{c_2}$ is the state derivative (acceleration in the constrained directions), u_c is the secondary auxiliary input which can be designed using the force sensor information to achieve a stable and robust performance of the robot force control. With the new feedback scheme, (8) becomes

$$\begin{bmatrix} \dot{x}_{c_1} \\ 0 \end{bmatrix} = \begin{bmatrix} x_{c_2} \\ u_c \end{bmatrix} \tag{9}$$

It can be seen that position and output force decoupling is achieved without any assumption.

In fact, equation (9) can be formulated as follows

$$E_c \dot{X}_c = A_c X_c + B_c u_c \tag{10}$$

where

$$E_c = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \; A_c = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \; B_c = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Since E_c is a singular matrix, equation (10) is a representation of Singular system [15] [18]. In contrast to the system in which E_c is a nonsingular matrix, a Singular system usually consists of a dynamic part and nondynamic constraints. The dynamics of a singular system is a combination of exponential motions and impulsive motions which result from applying arbitrary initial condition of discontinuous input to the system [15].

Since impulsive motions may cause damage to the robot system, they are undesirable for most control applications. Hence, the first objective of designing a controller at this point is to find a feedback u_c yielding a system that has no impulsive behavior. In addition, a unified controller structure for all of the three different modes is desired.

3 Unified Force-Referenced Controller

Designing a controller based on Singular System Theory can be put into the following steps. First of all, at the level of the decoupled manipulator, by selecting a unified control structure,

$$V = \ddot{\bar{y}}^d + k_v \left(\dot{\bar{y}}^d - \dot{y} \right) + k_p \left(\bar{y}^d - y \right) \tag{11}$$

the linear dynamic behaviors of positions and forces can be obtained.

Secondly, for each mode an unified action plan can be designed as

$$\begin{cases} \bar{y}^d = y^d + k_s \int (F^d - F)dt \\ \dot{\bar{y}}^d = \dot{y}^d + k_s(F^d - F) \\ \ddot{\bar{y}}^d = \ddot{y}^d + k_s(\dot{F}^d - \dot{F}) \end{cases} \tag{12}$$

where

$$\bar{y}^d = \begin{bmatrix} \bar{y}_u^d \\ \bar{y}_c^d \end{bmatrix}, \qquad y^d = \begin{bmatrix} y_u^d \\ y_c^d \end{bmatrix},$$

and

$$F = \begin{bmatrix} 0 \\ F_c \end{bmatrix}, \qquad F^d = \begin{bmatrix} 0 \\ F_c^d \end{bmatrix},$$

$$dim\ \bar{y}^d = dim\ \bar{y}_u^d + dim\ \bar{y}_c^d$$

$$dim\ y^d = dim\ y_u^d + dim\ y_c^d$$

and F^d, F, k_s represent the desired force, the actual force response and the force gain in the constrained directions. Let t_{sw} represent the time of the detection of the impact. The above action plan consists of two parts, motion plan and force plan.

The motion plan in unconstrained directions is :

$$\begin{cases} \bar{y}_u^d = y_u^d & \text{desired path} \\ \dot{\bar{y}}_u^d = \dot{y}_u^d & \text{desired velocity on the path} \\ \ddot{\bar{y}}_u^d = \ddot{y}_u^d & \text{desired acceleration on the path} \end{cases}$$

Therefore,

$$V_u = \ddot{y}_u^d + k_v\left(\dot{y}_u^d - \dot{y}_u\right) + k_p\left(y_u^d - y_u\right) \qquad \text{(mode I, II, III)} \tag{13}$$

The motion plan in constrained directions:

$$\begin{cases} \bar{y}_c^d = y_c^d & \text{desired path} \\ \dot{\bar{y}}_c^d = \dot{y}_c^d & \text{desired velocity on the path} \\ \ddot{\bar{y}}_c^d = \ddot{y}_c^d & \text{desired acceleration on the path} \end{cases} \qquad t < t_{sw} \tag{14}$$

or

$$\begin{cases} \bar{y}_c^d = y_c + k_s \int (F_c^d - F_c)dt \\ \dot{\bar{y}}_c^d = \dot{y}_c + k_s(F_c^d - F_c) \\ \ddot{\bar{y}}_c^d = \ddot{y}_c + k_s(\dot{F}_c^d - \dot{F}_c) \end{cases} \qquad t \geq t_{sw} \tag{15}$$

The force plan in constrained direction:

$$F_c^d = \begin{cases} 0 & t < t_{sw} \\ f_c^d & t \geq t_{sw} \end{cases}$$

Therefore,

$$V_c = \ddot{y}_c^d + k_v\left(\dot{y}_c^d - \dot{y}_c\right) + k_p\left(y_c^d - y_c\right) \qquad t < t_{sw} \tag{16}$$

$$V_c = \ddot{y}_c + k_s(\dot{f}_c^d - \dot{F}_c) + k_v k_s(f_c^d - F_c) + k_p k_s \int (f_c^d - F_c)dt \qquad t \geq t_{sw} \tag{17}$$

After defining $u_c = k_s(\dot{f}_c^d - \dot{F}_c) + k_v k_s(f_c^d - F_c) + k_p k_s \int (f_c^d - F_c)dt$ and substituting into equation (17), we have $V_c = \ddot{y}_c + u_c$ which is equivalent to equation (8). Therefore, u_c is the control command for the singular system (9). Clearly, u_c is a PID force controller. It only contains the dynamics of the force errors. As a matter of fact, the interaction force is a nondecreasing function of the environmental deformation. Hence u_c implicitly contains the position dynamics of the manipulator which is a dynamic constraint, in other words, u_c regularizes the system (9).

The error dynamics in unconstrained directions is $\ddot{e}_u + k_v \dot{e}_u + k_p e_u = 0$ where $e_u = \bar{y}_u^d - y_u$.

By applying the conventional state space pole placement method, desired transient and steady state response can be obtained for the three modes.

In the constrained directions, if we let $e_c = \bar{y}_c^d - y_c$, $e_f = f_c^d - F_c$, from equations (16) and (17) the error dynamics will be

$$\ddot{e}_c + k_v \dot{e}_c + k_p e_c = 0 \tag{18}$$

where

$$e_c = \begin{cases} {y_c}^d - y_c & t < t_{sw} \\ k_s \int e_f dt & t \geq t_{sw} \end{cases}$$

Consequently, the error dynamics in Mode I, II and III are

$$\begin{cases} (\ddot{y_c}^d - \ddot{y}_c) + k_v(\dot{y_c}^d - \dot{y}_c) + k_p({y_c}^d - y_c) = 0 & t < t_{sw} \quad \text{(mode I)} \\ k_s \dot{e}_f + k_v k_s e_f + k_p k_s \int e_f dt = 0 & t \geq t_{sw} \quad \text{(mode II, III)} \end{cases}$$

Therefore, by choosing k_v, k_p, desired transient and steady state force response can be obtained without knowing the environmental model.

In summary, on the detection of impact, the plan will switch from (14) to (15), the control purpose will switch from position control to force control, and the force error will go to zero asymptotically without large impact force and bouncing after switching.

4 Simulation and Experiment Results

The simulation of the transition and contact task for a PUMA 560 manipulator has been done on a SGI workstation with a graphical display in the Robotic Laboratory of Washington University. The motion planning for the simulation and experiment are based on event-based planning techniques [16]. The measurements of the joint accelerations and output forces in the constrained direction are required. The linear force model with a stiffness value of 1000.0 kg/m was used for the contact surface in this simulation. The surface location is set to be $Z = -0.430$m. Therefore, $\{(x\ y\ z) | x \in \mathbb{R}, y \in \mathbb{R}, z > -0.430\}$ and $\{(x\ y\ z) | x \in \mathbb{R},\ y \in \mathbb{R}, z \leq -0.430\}$ represent the free space and the constrained space respectively. The desired path for the approaching mode is a straight line from $A = (0.6\ 0,\ -0.3)$ to $B = (0.4,\ 0.3,\ -0.431)$ where B is the transition point in the constrained space. From B to C, a desired path on the surface is specified for the contact motion mode. The position of point C on the surface is $(x, y) = (0.7,\ 0)$. Figure 1 shows the force response for a desired force value of -1.0 kg. Figure 2 gives the force error profile. It can be seen that the error goes to zero quickly without bouncing. Thus, the impact force is effectively controlled by the proposed control scheme, and force tracking

is accomplished perfectly. Figure 3 and 4 show the time evolution of the positions and velocities respectively. Hence, the given control method results in smooth phase transition except a small jump during the impact mode.
The experiment of the above control algorithm was implemented on a 6-DOF PUMA 560 robot arm which was equipped with a FSA-3254 6-Axis Force/Torque Sensor. The force/torque was measured at a rate of 1000Hz, and the filtered force/torque was computed at a rate of 500Hz. The sampling rate for control and joint position/velocity measurement was 1000Hz. Due to the inaccuracy of the measurements of the acceleration and force derivative, some modification of the control scheme was made. The acceleration was multiplied by a factor less than 1, and the force derivative is replaced by the velocity. In the experiments the positions in the x, y and z directions and orientation (O,A,T) were commanded for free space motion. After the detection of contact only force was commanded in the z direction. The surface location is somewhere between $Z = -0.50$m and $Z = -0.60$m. Figure 5 shows the force response for a desired force value of -1.0 kg. Figure 6 gives the force error profile. The absolute value of force error is within 200grams when the end-effector moves along the surface. The experimental results are consistant with the simulation. Figure 7 and 8 show the time evolution of the positions and velocities.

5 Conclusions

A novel integrated planning and control method for robot phase transition control has been proposed. Combining Singular system theory and nonlinear feedback linearization, the dynamic model can be linearized and decoupled in the free motion mode, the constrained motion mode as well as the phase transition mode without assuming zero acceleration and zero velocity in the constrained directions, an assumption necessary for the implementation of many of the existing phase transition control schemes. Based on the proposed planning and control scheme, a stable phase transition and closed-loop dynamic force control has been devised, large impact forces are avoided with the environmental uncertainty. Moreover, the output force in constrained directions can be regulated. The advantage of the proposed scheme is further demonstrated by its unified structure which provides an efficient way for implementation in practical applications.

References

[1] H. Kazerooni, B. Waibel and S. Kim, On the Stability of Robot Compliant Motion Control: Theory and Experiments, Transaction of ASME, Journal of Dynamics Systems Measurements and Control, Vol. 112, pp. 417–425, 1990.

[2] N. Hogan, Impedance Control: An Approach to Manipulation: Part I - Theory and Part II - Implementation, J. Dynamical Systems, Meas. Control, Vol. 107, March 1985, pp. 1–16.

[3] S.A. Schneider, R.H. Cannon, Jr., Object Impedance Control for Cooperative Manipulation: Theory and Experimental Results, IEEE Trans. R& A, Vol. 8, No. 3, June 1992, pp. 383–395.

[4] M.W. Spong, M. Vidyasagar, Robot Dynamics and Control, Wiley and Sons, 1989.

[5] O. Khatib, A unified approach for motion and force control of robot manipulations. The operational space formulation, IEEE Journal of R& A, Vol. RA-3, No. 1, February 1987, pp. 43–53.

[6] J.K. Mills, D.M. Lokhorst, Control of Robotic Manipulators During General Task Execution: A Discontinuous Control Approach, Int. J. Rb. Res., Vol. 12, No. 2, April 1993, pp. 861–874.

[7] J. Hyde, M. Cutkosky, Contact Transition Control: An Experimental Study, Proceedings of International Conference on Robotics and Automation, 1993, pp. 363–368.

[8] R. Volpe, P. Khosla, A Theoretical and Experimental Investigation of Impact Control for Manipulators, The International Journal of Robotics Research, Vol. 12, No. 4, August 1993, pp. 352–365.

[9] G.T. Marth, T.J. Tarn, A.K. Bejczy, Stable Phase Transition Control for Robot Arm Motion, Proceedings of International Conference on Robotics and Automation, 1993, pp. 355–362.

[10] Y.F. Zheng, H. Hemami, Mathematical Modeling of a Robot Collision with its Environment, Journal of Robotics Systems, Vol. 2, No. 3, 1985, pp. 289–230.

[11] Y. Wang and M. Mason, "Modeling Impact Dynamics for Robotic Operations," Proc. of the IEEE International Conference on Robotics and Automation, April 1987.

[12] K. Youcef-Toumi and D.A. Gutz, Impact and force control, Proc. IEEE ICRA, 1989, pp. 410–416.

[13] T.J. Tarn, A.K. Bejczy, A. Isidori and Y.L. Chen, "Nonlinear feedback in robot arm control," in Proc. 23rd IEEE Conf. Robotics Automat. (Las Vegas, NV), 1984.

[14] Robert R.Y. Zhen, Andrew A. Goldenberg, Robust Position and Force Control of Robots Using Sliding Mode, Proc. of the IEEE International Conference on Robotics and Automation, Vol. 1, 1994, pp. 623–628.

[15] Z. Zhou, Feedback Synthesis of Singular Systems— A Geometric Approach, Ph.D. Dissertation, Dec. 1984.

[16] T.J. Tarn, N. Xi, A.K. Bejczy, Motion Planning in Phase Space for Intelligent Robot Arm Control, Proc. IEEE/RSJ Int. Conf. Robots and Systems, Raleigh, 1992, pp. 1507–1514.

[17] P. Akella, V. Parra-Vega, S. Arimoto and K. Tanie, Discontinuous Model-based Adaptive Control for Robots Executing Free and Constrained Tasks, Proc. of the IEEE International Conference on Robotics and Automation, Vol. 4, 1994, pp. 3000-3007.

[18] G.C. Verghese, B.C. Levy, T. Kailath, A Generalized State-Space for Singular Systems, IEEE Transactions on Automatic Control, Vol. AC-26, No. 4, August 1981.

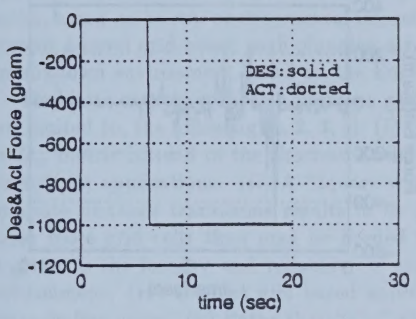


Figure 1. Force response

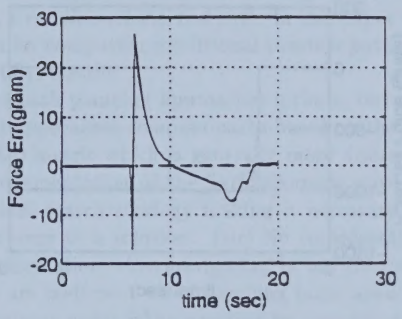


Figure 2. Force error(DES-ACT)

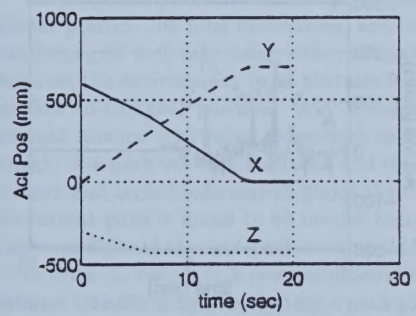


Figure 3. Position response

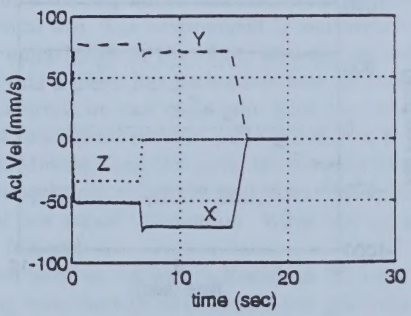


Figure 4. Velocity response

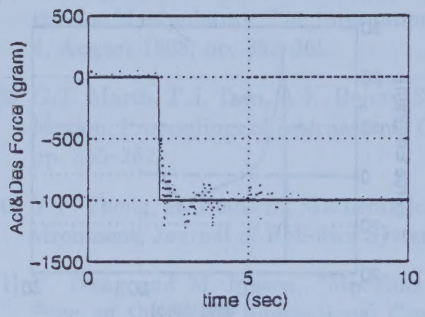


Figure 5. Desired force

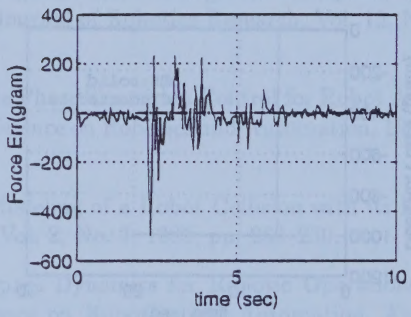


Figure 6. Force error(Des-Act)

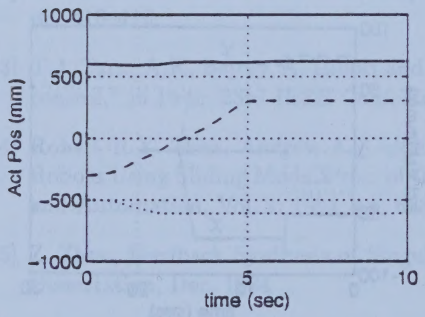


Figure 7. Position response

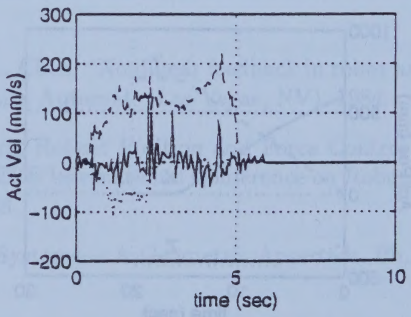


Figure 8. Velocity response

PLANNING CONDITIONAL SHORTEST PATHS IN AN UNKNOWN ENVIRONMENT

Robert J. Szczerba *Danny Z. Chen* *John J. Uhran Jr.*

Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, Indiana 46556

A realistic and challenging problem that has been generating considerable attention in recent years is navigation through an unknown environment, where path planning is implemented "on the fly" as new obstacle information becomes available through external sensors. Obviously, without any *a priori* knowledge of an environment there is no way to determine a true shortest path to a desired destination. We call the path whose length is the shortest among all possible paths, based on known environmental information, a *conditional shortest path.* In this paper we present a novel grid-based path planning algorithm for computing conditional shortest paths in an unknown environment based on the Euclidean (L_2) metric.

Problems encountered with previous grid-based path planning approaches include, but are not limited to, the following [1, 2, 3, 4]: (*i*) Current approaches compute paths based on the L_1 or L_∞ metric instead of the shortest Euclidean (L_2) metric which is generally more desirable for robotic applications. (*ii*) A "square wave" implementation of the path planning wave to compute distance transforms results in an inefficient search strategy making it necessary to cover more grid cells than may be needed to converge to a solution. (*iii*) No consideration is given to the locality and reliability of sensor information when navigating in an unknown environment. (*iv*) Current grid-based approaches are inefficient when handling large areas of obstacle-free space. (*v*) Paths that "clip" obstacles may occur when considering movement to diagonally-adjacent grid cells. Our approach removes these limitations and provides both an accurate and efficient solution to this problem.

We propose a *generate-and-test paradigm* in which the only initial information is the robot's current position, its final destination, and its physical size. The environment is represented by a uniform grid with cells being either navigable or unnavigable by the robot. Initially, the robot perceives the environment to be obstacle-free and that a direct line path exists from its current position to the goal position. Any subsequent information can come only from the robot's external sensors. The robot follows the calculated conditional shortest path (initially, it is the straight line path between the robot and the goal). Moving along this path, the robot continues to sense and record information about the environment until either the goal is reached or until the current path is found to be invalid because of new sensor information. When this occurs, a new conditional shortest path is computed and followed.

In order to calculate a new conditional shortest path we compute a Euclidean (L_2) metric distance transform by propagating a path planning wave through the obstacle-free grid cells to assign *effective costs* to different locations in the environment. The wave propagates radially outward from the robot's position, bending around obstacles, until it intersects the *goal-front.* The goal-front is defined as the maximum circular area centered around the goal in which there has been no detected obstacles. The radius of the goal-front shrinks as new environmental information becomes available. The goal-front takes advantage of the fact that in an unknown environment most of the known obstacles will be closer to the robot than to the goal, because nearer obstacles obstruct obstacles further away. Once the path planning wave intersects any point on the radius of the goal-front, the calculation of the rest of the path is trivial.

Our method for propagating the wave and computing the distance transform is based on a generalized breadth-first approach. Note that a normal breadth-first approach that is typically used would not work here because the edge weights of the underlying graph vary. We are assuming that the effective cost to move to an edge-adjacent grid cell is 1 while the cost to move to a diagonally-adjacent grid cell is $\sqrt{2}$. The procedure performs a number of iterations.

It uses a control parameter, called *wave_val*, to decide which grid cells the path planning wave can cover in each iteration. Precisely, the value of *wave_val* at iteration i is equal to i, and the wave-front, during iteration i, covers all the cells such that the L_2 lengths of the paths from the robot's current position to those cells are $> i-1$ (the value of *wave_val* at iteration $i-1$) but $\leq i$ (the current value of *wave_val*). The value of *wave_val* is increased by 1 per iteration. This produces a "circular" path planning wave as opposed to the "square" waves used previously.

If the wave cannot propagate further and if it does not yet intersect the goal-front, then the procedure stops, knowing that the goal is not reachable. So without loss of generality, assume that the wave-front intersects the goal-front, and that the shortest path goes through the grid cell called *sub-goal* on the boundary of the goal-front. Obviously, the shortest path we seek consists of the shortest path between the robot's current position and the *sub-goal*, as well as the straight line path between the *sub-goal* and the goal. The algorithm finds the shortest path between the robot and *sub-goal* by tracing the path starting at *sub-goal* back to the robot's current position, based on the grid cell values computed during the wave propagation process. This tracing procedure finds the next grid cell on the path by identifying the adjacent grid cell with the lowest total cost. As the robot moves along this calculated path, it continually senses and maps new sensor information to the grid. The radius of the goal-front is reduced as new obstacle information closer to the goal becomes available.

Our approach takes into account the reliability and locality of sensor information by computing the distance transform from the robot to the goal (instead of the customary goal-to-robot propagation). This robot-to-goal propagation takes advantage of the fact that in an unknown environment most of the known obstacle information will be closer to the robot than to the goal. This allows the introduction of a goal-front concept which significantly reduces the number of calculations needed to converge to a shortest path in environments with large areas of free space. The computational savings achieved with the goal-front is especially evident when dealing with a cluttered, or maze-like, environment where closer obstacles obstruct the robot's view of obstacles farther away. Furthermore, the proposed algorithm performs simple local tests which eliminate the possibility of movement to cells which could clip obstacles, but still includes diagonally-adjacent grid cells which may lead to a legitimate shortest path. Because we are using a distance transform approach we can easily detect if the goal is reachable or not from the robot's current position. Our approach effectively implements a generalized breadth-first strategy which can handle the different edge weights associated with diagonal, horizontal, and vertical movements. The algorithm allows movement in eight directions and finds a shortest path based on the Euclidean (L_2) metric as opposed to the L_1 or L_∞ metrics previously used. A linear-time wave propagation (based on the number of cells covered) is implemented using a circular wave-front resulting in the ability to cover only those cells essential for finding the shortest path. Previous work using a square wave will not work here. Furthermore, our approach is easily extendable to multiple goals and multiple robots in an environment. A detailed description of our algorithm along with a complete discussion of its advantages, disadvantages and simulation results are presented in the full paper.

References

[1] Y. Hwang and N. Ahuja. Gross motion planning - a survey. *ACM Computing Surveys*, 24(3):219–291, September 1992.

[2] R.A. Jarvis and J.C. Byrne. Robot navigation: Touching, seeing and knowing. *Proceedings of the 1st Australian Conference on Artificial Intelligence*, November 1986.

[3] J. Latombe. *Robot Motion Planning.* Kluwer, Boston, MA, 1991.

[4] K. Trovato. Differential A*: An adaptive search method illustrated with robot path planning for moving obstacles and goals, and an uncertain environment. *International Journal of Pattern Recognition*, 4(2):245–268, 1990.

H∞ CONTROL OF A SINGLE FLEXIBLE LINK ROBOT MANIPULATOR USING A STATE-SPACE APPROACH

Sylvain Dindy-Bolongo and Edwin Sawan

The Wichita State University
Department of Electrical Engineering
1845 Fairmount
Wichita, KS, 67208

Abstract

This paper treats the problem of control of a single flexible link manipulator based on the H∞ control theory. The dynamic model is derived using the Lagrangian formulation. This model is based on a set of cantilever modes. The controller is derived using the state space formulation of the standard H∞ problem: i.e for a given number $\gamma \geq 0$, *the controller must be such that the H∞ norm of the closed-loop transfer matrix is strictly less than* γ. *The performances of this controller are tested through simulation.*

1.0 INTRODUCTION

Today, robots are used for various purposes in different industries. The main disadvantages of the robots currently used are their very heavy construction, as well as their relatively slow speed. In recent years, improvements in electric motor technology, coupled with new designs, such as direct-drives arms, have led to a rapid increase in the speed and load-carrying capabilities of the manipulators. However, this has meant that the flexibility of the nominally rigid links has become increasingly significant. The ratio of the payload to weight of an industrial robot now ranges from 1:10 to 1:30 and less. Compared with that of man, it is extremely low: an individual is able to have a ratio of 1:3 or more[4,5]. If the controller could compensate for link flexure, it would, in principle, be possible to increase the ratio of load to arm weight greatly.
There are also other potential advantages arising from the use of flexible link[5]:
- Lower energy consumption; lighter links have lower inertia and; therefore, require less power to produce the same load carrying capacity.
- Smaller actuator required.
- Safer operation due to the reduced inertia.
- Compliant structure; flexible links introduce mechanical compliance into the robot structure. This is useful for delicate assembly operations.

An important aspect of flexible arm control that has received little attention is the interaction between the gross dynamics and the deformational dynamics of the links. As the arm configuration changes, the spacial boundary conditions of the links change thereby modifying their characteristic frequencies and modes. This has important implications for the control design in which the performance of a controller designed on the basis of a fixed linear model of the dynamics may be seriously degraded. Faced with this problem, the control engineer has the following options:
- To develop a controller based on a more general model of the system.
- To use robust control schemes which can maintain a satisfactory level of performance despite change in the arm dynamics.
- To use adaptive control schemes which modify the controller in line with changes in the

system.
The work presented here falls into the first group. Control strategies in this category can be divided into classical control design and modern control design..

. *Classical Control Design:*
Early solutions to the control of the flexible robot use classical control design. This kind of design includes the feedback control of the flexible state variables, as well as the joint variables. This control schemes sometimes uses proportional and rate feedback to achieve the desired closed loop pole position.
. *Modern Control:*
In this group there are two approaches that have proved to be popular: the pole placement and the optimal regulation formulation[9,10,11]. The Pole Placement method involves specifying the desired closed-loop performace in terms of the closed-loop pole positions. The feedback gains, necessary to yield the desired pole positions can then be calculated. Some of the states may have to be reconsructed via an observer. The Optimal Control method finds the feedback gain such that a given performance index is minimized in addition to satisfying the stability requirement.
Central to the optimal state feedback regulator is the definition of a suitable cost function. The commonly chosen performance is based on the H2 norm. There is another norm that is sometimes used in optimal control theory: the H∞ norm. In the past, most of the results of the H∞ control theory were obtained in the frequency domain, but the computations involved in the design of such controller are sometimes expensive. Recently, considerable interest has been focused on the state space solution which reduced the general H∞ optimal control to the solution of two algebraic Riccati equations [1]. Lin[2] uses a combination of the differential geometric structure algorithm and a state feedback H∞ optimal control theory to design a controller of a double link flexible manipulator.
In this work, we consider a different problem: Synthesizing a controller for a single link flexible manipulator using the state feedback H∞ control theory.
This paper is organized as follows: In section 2, the dynamic model of the flexible manipulator based on Lagrangian formulation is presented. In section 3, the design of a controller for the single link case is presented. The simulation results are presented in section 4. Some conclusions are given in section 5.

2.0 DYNAMIC MODEL

Consider a uniform, slender beam connected via a rigid hub to the armature of an electric motor and having a tip mass (payload) with inertia, see fig 1. The beam is assumed to be initially straight and to satisfy the assumption of the Euler-Bernouilly beam theory. Note that in what follows ; **x** is the distance along the length of the beam; **l** is the length of the beam δ is the thickness of the beam; **m** is the mass per unit length; **EI** is the flexural rigidity of the beam; **r** is the hub radius; $\mathbf{I_h}$ is the hub inertia; $\mathbf{M_t}$ is the payload mass; $\mathbf{I_t}$ is the payload inertia; $\mathbf{I_b}$ is the inertia of the beam about the motor armature.
The displacement of joints along the deformed profile of the beam is described in terms of the radial and circumferential coordinates x and y and is related to the angles of rotation of the hub θ and the flexural displacement of the beam as follows:

$$y(x,t)=w(x,t)+(x+r)\theta(t) \qquad (1)$$

In accordance with the Euler-Bernouilly beam theory, the beam deflection w(x,t) is assumed to be small; therefore, the kinetic energy T_k , the potential energy V_e , are given as follows:

$$T_k=\frac{1}{2}\{I_h\dot{\theta}^2+\int_0^l \dot{y}(x,t)^2 m dx+M_p\dot{y}(l,t)^2+I_p\dot{y}(l,t)^2\} \quad (2)$$

$$V_e=\frac{1}{2}\int_0^l EIy''(x,t)^2 dx \quad (3)$$

in which "." denotes the derivative with respect to time and ′ denotes the derivative with respect to the distance. The displacement can be expanded using the assumed-mode approach as follows:

$$y(x,t)=\sum_0^\infty \phi_i(x)q_i(t) \quad (4)$$

in which q_i is purely a function of time and includes an arbitrary multiplicative constant and ϕ_i , the mode shape, is purely a function of the displacement along the beam.
We can choose ϕ_i as the natural mode or any complete set of admissible functions. Natural modes vary with gross dynamics and configuration of the arm; therefore, requiring frequent recalculations[4]. Because of this, we choose ϕ_i as the cantilever mode. The mode shape of a cantilever beam can be expressed as:

$$\phi_i(x)=A(\cosh(\lambda_i x)-\cos(\lambda_i x)-k_i(\sinh(\lambda_i x)-\sin(\lambda_i x)) \quad (5)$$

in which:

$$k_i=(\cos(\lambda_i l)+\cosh(\lambda_i l)/(\sin(\lambda_i l)+\sinh(\lambda_i l) \quad (6)$$

and λ_i are root of :

$$1+\cos(\lambda_i l)\cosh(\lambda_i l)=0 \quad (7)$$

It can be shown that these modes satisfy certain orthogonality relationships[4]. In terms of the cantilever modes, the displacement of any point on the beam is represented by the summation:

$$y(x,t)=\sum_1^\infty \phi_i(x)q_i(t)+(x+r)\theta \quad (8)$$

substituting (8) into (2) and (3) yields:

$$T_k=\frac{1}{2}\{(I_h+I_b)\theta^2+4ml^3\theta\sum_1^\infty q_i/\lambda_i^2+ml^3\sum_1^\infty q_i^2\} \quad (9)$$

and:

$$V_e=\frac{1}{2}ml^3\sum_1^{\infty}\omega_i^2 q_i^2 \tag{10}$$

Define the Lagrangian to be:

$$L=T_k-V_e \tag{11}$$

The dynamic equation is given as :

$$\partial L/\partial q_i-d/dt(\partial L/\partial \dot{q}_i)=\tau \tag{12}$$

for i=0,1, ... and τ motor torque . Equation (12) yields an infinite set of coupled second-order differential equations. Retaining the first n elastic modes and writting these equations in matrix form, we obtain:

$$M\ddot{q}+K_c q=Q \tag{13}$$

in which:

$$M=\begin{bmatrix} I_T & 2\frac{ml^3}{\lambda_1^2} & . & . & 2\frac{ml^3}{\lambda_n^2} \\ 2\frac{ml^3}{\lambda_1^2} & ml^3 & . & . & 0 \\ . & . & . & . & . \\ 2\frac{ml^3}{\lambda_n^2} & 0 & . & . & ml^3 \end{bmatrix} \tag{14}$$

$$q=\begin{bmatrix}\theta\\ q_1\\ .\\ .\\ .\\ q_n\end{bmatrix} \tag{15}$$

$$K_c=\begin{bmatrix}0 & 0 & 0\\ 0 & ml^3\omega_1^2 & 0\\ 0 & 0 & ml^3\omega_n^2\end{bmatrix} \tag{16}$$

$$Q=\begin{bmatrix}\tau\\ 0\\ .\\ .\\ 0\end{bmatrix} \tag{17}$$

Equation (13) can be transformed into the following state space form:

$$\dot{x}=Ax+B\tau \tag{18}$$

in which :

$$A=\begin{bmatrix}0 & -M^{-}1K_c\\ I & 0\end{bmatrix} \tag{19}$$

and:

$$B=\frac{1}{I_t}\begin{bmatrix}1\\ \phi_1'\\ \cdot\\ \cdot\\ \cdot\\ \phi_n'\\ 0\\ 0\\ \cdot\\ \cdot\\ \cdot\\ 0\end{bmatrix} \tag{20}$$

$$x=\begin{bmatrix}\dot{\theta}_0\\ \dot{q}_1\\ \cdot\\ \cdot\\ \cdot\\ \dot{q}_n\\ q_0\\ \cdot\\ \cdot\\ \cdot\\ q_n\end{bmatrix} \tag{21}$$

3.0 H∞ CONTROL DESIGN

Define τ =w+u where w some disturbances and u the control input. Let $z=C_1x$ be the output and y the measured variable.
The basic block diagram used for the synthesis of the H∞ state feedback controller is shown in fig 2. in which the realization of the transfer matrix G is in the form:

$$G(s)=\left[\begin{array}{c|cc} A & B & B \\ \hline C_1 & 0 & 0 \\ I & 0 & 0 \end{array}\right] \tag{22}$$

where (A,B), (I,A) are stabilizable and (C_1,A) detectable. The D matrix (=0) does not meet the stronger assumption of [1], but it can be shown that this assumption can be relaxed in the suboptimal application to flexible robot[2], Simulation results also confirm this fact.

Now, let us consider the following suboptimal H∞ control problem: find all admissible controller K∞ such that:

$$\| T_{zw} \|_\infty \leq \gamma \tag{23}$$

in which T_{zw} is the closed-loop transfer matrix from w to z and γ is a design parameter. We suppose that the full state is avalable; therefore, the central controller is simply a gain matrix K∞ compute from the solution to the following ARE (Algebraic Riccati Equation) :

$$A^T X_\infty + X_\infty A + X_\infty(\gamma^{-2}-1)BB^T X_\infty + C_1^T C_1 = 0 \tag{24}$$

associated with the following Hamiltonian matrix[1]:

$$H_\infty = \begin{bmatrix} A & -(1-\gamma^{-2})BB^T \\ -C^T C & -A^T \end{bmatrix} \tag{25}$$

The required control law is given as u=-K∞x in which $K_\infty=-B^T X_\infty$ and X_∞ is the non-negative definite stabilizing solution to the above H∞ control ARE.

4.0 SIMULATION EXPERIMENT

Theoretically, the number of generalized coordinates is infinite (n= ∞). The model was truncated so that the state variable has dimension 6. The simulation of the performance of this controller is perfomed using MATLAB. And the following parameters were used $m=1 kg.m^{-1}$, $EI=1 Nm^2$, l=0.5, $\omega_{c1}=10.7, \omega_{c2}=40.7$, $I_t=1.0$. The controller gain K∞ for $\gamma=1.7$ is computed; the step response in terms of the rigid mode history $\theta(t)$, the flexible mode histories $q_1(t)$, $q_2(t)$ are also computed. The comparison of the H∞ design with the LQR (Linear Quadrtic Regulator) problem with $Q= C_1^T C_1$ and R=I (identity matrix) is performed. Note that the LQR regulator corresponds to the case in which $\gamma= \infty$ and is associated with the following Hamiltonian matrice:

$$H2=\begin{bmatrix} A & -B_2B_2^T \\ -C_1^TC_1 & -A_T \end{bmatrix} \quad (26)$$

The results of the simulation are shown in fig 3, 4, 5, 6, 7. We can see that the H∞ control with γ =1.7 has better steady state errors than the LQR case. Moreover, the flexible mode are damped out quickly.
The fig 7 shows the plot of the maximum singular value of the closed-loop transfer matrix as functions of frequencies.The results indicate that the maximum singular value in the H∞ control design decreases for this range of frequences and this value of γ.

5.0 CONCLUSION

The effect of link flexibility on the overall dynamic performance of a flexible robotic system are very significant. They are sources of trajectory tracking errors and undesirable oscillations of the end effector. In this paper, we have presented a feedback design procedure based on the H∞ control theory. Moreover we tested the proposed control strategy via computer simulation on a single flexible link manipulator. The simulation results demonstrate that a significant reduction in vibration can be achieved using the H∞ control.

6.0 REFERENCES

[1] Doyle J.C., K. Glover, P. Khargonekar, B.A. Francis " State-space Solution to Standard H2 and H∞ Control Problems".
IEEE Transaction on Automatic Control, No 3, August 1989.

[2] Lin L.C " State Feedback H∞ Control of Manipulator with Flexible Joints and Links"
Proceeding of the 1991 IEEE International Confererence On Robotic and Automation Sacramento California April, 1991.

[3] Desoyer K., P. Kopacek, P. Lugner and I. Troch " Flexible Robots- A Survey"
IFAC Theory of Robots Vienna Austria, 1986.

[4] Fraser A.R and R.W Daniel " Perturbation Techniques for Flexible Manipulators".
Kluwer Academic Publishers ,1991.

[5] Troch I. and P. Kopacek " Control concepts and Algorithms for Flexible Robots- An Expository Survey".
IFAC Robot Control he Federal Republic of Germany, 1988.

[6] Kanoh H. " On Properties of Transfer Functions of a Flexible Structure".
IFAC Control of Distributed Parameter Systems Perpignan France, 1989.

[7] Lewis F.L and Vandegrift " Flexible Robot Arm Control by a Feedback Linearization/Singular perturbation Approach".
Proceeding of IEEE , 1993.

[8] Mills J.K. " Stability and Control Aspects of Flexible Link Robot Manipulator During Constrained Motion Tasks"
Journal of Robotics Systems 9(7) pp933-953, 1992

[9] Book W.J. and Majette M. " Controller design For Flexible Distributed Parameter Mechanical Arms via Combined State-Space and Frequency Domain Techniques".
Trans. ASME J.Dyn.Sys. Meas. and Control., pp424-431, Dec 1975.

[10] Truckenbrodt A. " Truncation Problems in the Dynamics and Control of Flexible Mechanical System".
Proc. IFAC 5th Triennal World Congress On Control Sciences and Technology Kyoto Japan, 1981.

[11] Karkkainen P. " Compensation of Manipulator Flexibility Effects By Modal Space Techniques"
Proc. of IEEE Int Conf on Robotics and Automation St-Louis USA March 25-28th ,1981.

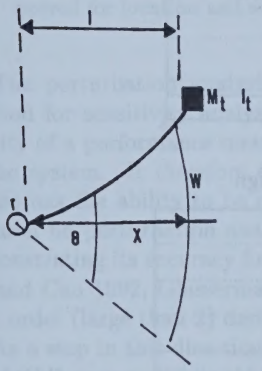


fig 1)Single link arm model

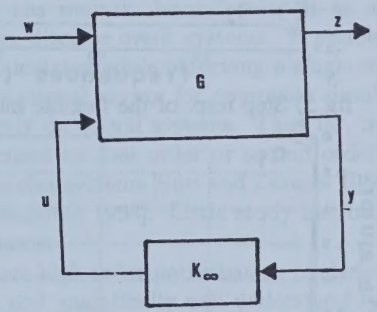


fig 2)Block diagram for the H∞ control

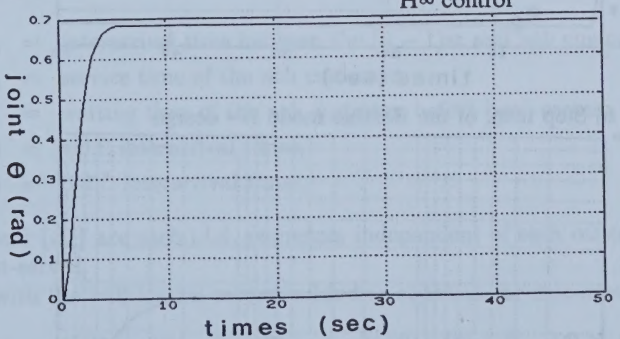


fig 3)Step resp. of the rigide mode θ, H2 design

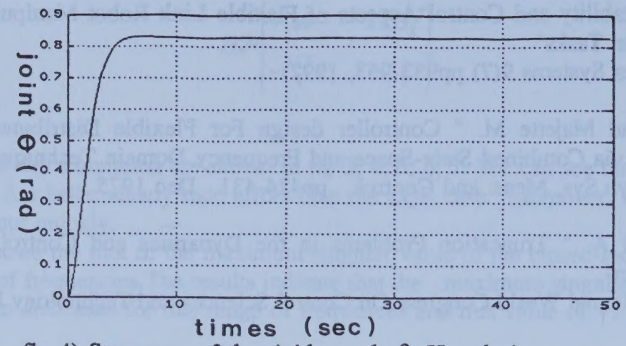


fig 4) Step resp. of the rigide mode θ, H∞ design

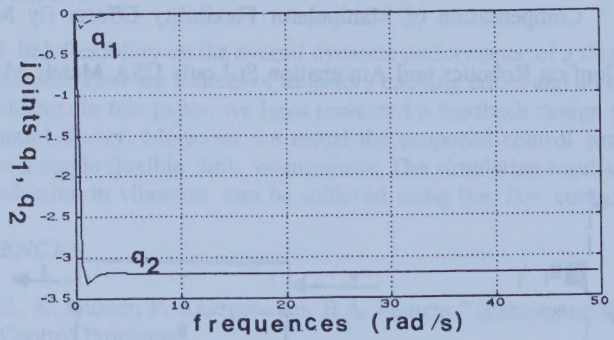


fig 5) Step resp. of the flexible mode, H2 design

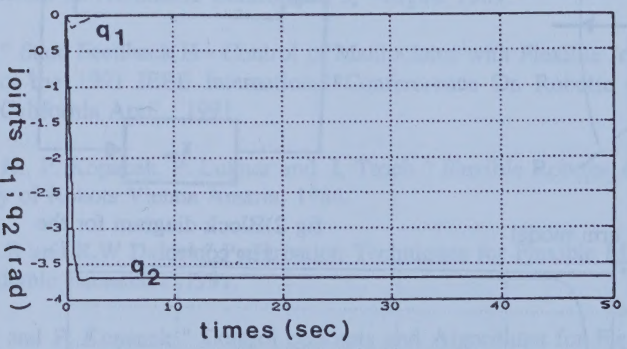


fig 6) Step resp. of the flexible mode H∞ design

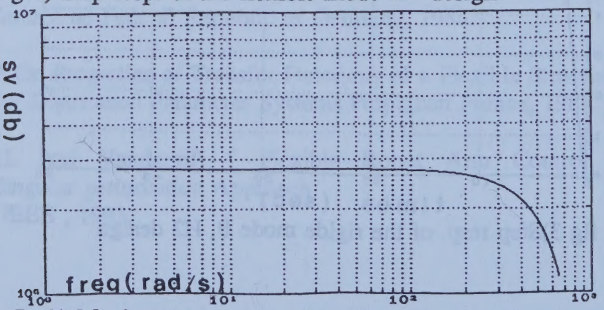


fig 7) Maximum singular value H∞ design

On the Consistency of High Order Derivative Perturbation Analysis Estimators for the M/G/1 Queue

Leyuan Shi
Department of Industrial Engineering
University of Wisconsin-Madison
Madison, WI 53706

Abstract

In this paper, high order derivative estimators of mean steady-state waiting time for the M/G/1 queue is derived. Strongly consistency of the estimator formula is proved for location and scale parameters of the service time distribution.

The perturbation analysis technique has recently been developed as an effective method for sensitivity analysis of complex discrete event systems. It enables the sensitivity of a performance measure to be calculated while observing a single sample path of the system. It therefore offers computational saving for computer simulation, and it also has the ability to be applied directly on actual systems. Thus far, much of the research on perturbation analysis has focused on first order or second order derivative demonstrating its accuracy for various complex systems [Suri and Zazanis 1988, Fu 1989, Ho and Cao 1992, Glasserman 1991, Cassandras 1993]. Little study has been done on high order (large than 2) derivative estimators.

As a step in this direction we study here high order perturbation analysis applied to an M/G/1 queue. While this is a simple and analytically well understood system from queueing theory, it nevertheless is nontrivial.

Define

$$\begin{aligned} A_n &= \text{interarrival time between the } (n-1)\text{st and } n\text{th customers,} \\ S_n &= \text{service time of the } n\text{th customer,} \\ W_n &= \text{waiting time of the } n\text{th customer before been served,} \\ F(\cdot) &= \text{c.d.f. interarrival times,} \\ f(\cdot) &= \text{p.d.f. interarrival times,} \end{aligned}$$

where $\{A_n\}$ and $\{S_n\}$ are each i.i.d. sequences independent of each other and service is first-come first-served.

We start with the well-known recursive Lindley equation for customer waiting time:

$$W_n(\theta) = \begin{cases} W_{n-1}(\theta) + S_{n-1}(\theta) - A_n & \text{if } W_{n-1}(\theta) + S_{n-1}(\theta) \geq A_n; \\ 0 & \text{if } W_{n-1}(\theta) + S_{n-1}(\theta) < A_n. \end{cases} \quad (1)$$

To derive the high order derivative estimators, we use a different way to represent Eq.(1). To do so, we first introduce the step function and its derivative δ- function in a very informal way *.

Define

$$H(x - g(\theta)) = \begin{cases} 1 & \text{if } g(\theta) \geq x; \\ 0 & \text{if } g(\theta) < x \end{cases} \tag{2}$$

where $g(\theta)$ is a continuous and differentiable function of θ.

Let

$$\frac{dH(x - g(\theta))}{d\theta} = \delta(x - g(\theta))\frac{dg(\theta)}{d\theta} \tag{3}$$

and δ-function satisfies

$$\int_0^\infty \delta(x - g(\theta))f(x)dx = f(g(\theta)) \tag{4}$$

where $f(x)$ is a continuous function of x.

Using the step function, given W_{n-1} and S_{n-1}, let

$$H(A_n - (W_{n-1} + S_{n-1})) = \begin{cases} 1 & \text{if } W_{n-1} + S_{n-1}) \geq A_n; \\ 0 & \text{if } W_{n-1} + S_{n-1} < A_n, \end{cases} \tag{5}$$

then the Lindely Eq.(1) can be represented as

$$W_n = (W_{n-1} + S_{n-1} - A_n)H(A_n - (W_{n-1} + S_{n-1})) \tag{6}$$

Taking derivative on Eq.(6) and note that $H(A_n - (W_{n-1} + S_{n-1}))$ is also a function of θ, we have

$$\begin{aligned} \frac{dW_n}{d\theta} = & \left(\frac{dW_{n-1}}{d\theta} + \frac{dS_{n-1}}{d\theta}\right) H(A_n - (W_{n-1} + S_{n-1})) \\ & + (W_{n-1} + S_{n-1} - A_n)\delta(A_n - (W_{n-1} + S_{n-1}))\left(\frac{dW_{n-1}}{d\theta} + \frac{dS_{n-1}}{d\theta}\right) \end{aligned} \tag{7}$$

Since the second term in above equation satisfies

$$\int_0^\infty (W_{n-1} + S_{n-1} - A_n)\delta(A_n - (W_{n-1} + S_{n-1}))\left(\frac{dW_{n-1}}{d\theta} + \frac{dS_{n-1}}{d\theta}\right) dF(A_n) = 0 \tag{8}$$

This means under the expectation

$$\frac{dW_n}{d\theta} = \left(\frac{dW_{n-1}}{d\theta} + \frac{dS_{n-1}}{d\theta}\right) H(A_n - (W_{n-1} + S_{n-1})). \tag{9}$$

Formula (9) is exactly the same as IPA estimator as seen in the PA literature.

For simplicity, in what follows, we use H and δ to represent $H(A_n - (W_{n-1} + S_{n-1}))$ and $\delta(A_n - (W_{n-1} + S_{n-1}))$, respectively. Using the same way, we can derive the second derivative estimator formula:

$$\frac{d^2W_n}{d\theta^2} = \left(\frac{d^2W_{n-1}}{d\theta^2} + \frac{d^2S_{n-1}}{d\theta^2}\right) H + \delta(A_n - (W_{n-1} + S_{n-1}))\left(\frac{dW_{n-1}}{d\theta} + \frac{dS_{n-1}}{d\theta}\right)^2 \tag{10}$$

*Details see [Shi 1994]

In order to calculate sample path derivative, we need to remove the δ-function. Since

$$\begin{aligned}
&\int_0^\infty \delta(A_n - (W_{n-1} + S_{n-1})) \left(\frac{dW_{n-1}}{d\theta} + \frac{dS_{n-1}}{d\theta}\right)^2 dF(A_n) \\
&= f(W_{n-1} + S_{n-1}) \left(\frac{dW_{n-1}}{d\theta} + \frac{dS_{n-1}}{d\theta}\right)^2 \\
&= \int_0^\infty \frac{f(W_{n-1} + S_{n-1})}{1 - F(W_{n-1} + S_{n-1})} \left(\frac{dW_{n-1}}{d\theta} + \frac{dS_{n-1}}{d\theta}\right)^2 \hat{H} dF(A_n) \\
&= \int_0^\infty \lambda \left(\frac{dW_{n-1}}{d\theta} + \frac{dS_{n-1}}{d\theta}\right)^2 \hat{H} dF(A_n)
\end{aligned}$$

where $\hat{H} = 1 - H$ and $\frac{f(W_{n-1}+S_{n-1})}{1-F(W_{n-1}+S_{n-1})} = \lambda$ because of Poisson arrival.

Hence, (10) is equal in expectation to

$$\frac{d^2W_n}{d\theta^2} = \left(\frac{d^2W_{n-1}}{d\theta^2} + \frac{d^2S_{n-1}}{d\theta^2}\right) H + \lambda \left(\frac{dW_{n-1}}{d\theta} + \frac{dS_{n-1}}{d\theta}\right)^2 \hat{H} \tag{11}$$

This again is coincident with the SPA second derivative estimator.

For deriving the third derivative estimator, we need to point out that the second term in Eq (11) is obtained under the condition that $A_n > W_{n-1} + S_{n-1}$. Which means $\hat{H} = 1$ and $d\hat{H}/d\theta = 0$. Therefore we have the third derivative estimator:

$$\begin{aligned}
\frac{d^3W_n}{d\theta^3} &= \left(\frac{d^3W_{n-1}}{d\theta^3} + \frac{d^3S_{n-1}}{d\theta^3}\right) H(A_n - (W_{n-1} + S_{n-1})) \\
&+ \delta(A_n - (W_{n-1} + S_{n-1})) \left(\frac{dW_{n-1}}{d\theta} + \frac{dS_{n-1}}{d\theta}\right) \left(\frac{dW_{n-1}^2}{d\theta^2} + \frac{dS_{n-1}^2}{d\theta^2}\right) \\
&+ 2\lambda \left(\frac{dW_{n-1}}{d\theta} + \frac{dS_{n-1}}{d\theta}\right) \left(\frac{dW_{n-1}^2}{d\theta^2} + \frac{dS_{n-1}^2}{d\theta^2}\right) \hat{H} \\
&= \left(\frac{d^3W_{n-1}}{d\theta^3} + \frac{d^3S_{n-1}}{d\theta^3}\right) H + 3\lambda \left(\frac{dW_{n-1}}{d\theta} + \frac{dS_{n-1}}{d\theta}\right) \left(\frac{dW_{n-1}^2}{d\theta^2} + \frac{dS_{n-1}^2}{d\theta^2}\right) \hat{H}
\end{aligned}$$

Again, the above equation should be interpreted in expectation because

$$\begin{aligned}
&\int_0^\infty \delta(A_n - (W_{n-1} + S_{n-1})) \left(\frac{dW_{n-1}}{d\theta} + \frac{dS_{n-1}}{d\theta}\right) \left(\frac{dW_{n-1}^2}{d\theta^2} + \frac{dS_{n-1}^2}{d\theta^2}\right) dF(A_n) \\
&= \int_0^\infty \lambda \left(\frac{dW_{n-1}}{d\theta} + \frac{dS_{n-1}}{d\theta}\right) \left(\frac{dW_{n-1}^2}{d\theta^2} + \frac{dS_{n-1}^2}{d\theta^2}\right) \hat{H} dF(A_n)
\end{aligned}$$

Clearly the third derivative can also be represented as

$$\frac{d^3W_n}{d\theta^3} = \begin{cases} \frac{d^3W_{n-1}}{d\theta^3} + \frac{d^3S_{n-1}}{d\theta^3} & \text{if } W_{n-1} + S_{n-1} \geq A_n; \\ 3\lambda(\frac{d^2W_{n-1}}{d\theta^2} + \frac{d^2S_{n-1}}{d\theta^2})(\frac{dW_{n-1}}{d\theta} + \frac{dS_{n-1}}{d\theta}) & \text{if } W_{n-1} + S_{n-1} < A_n, \end{cases} . \tag{12}$$

Under the assumption that the system regenerates infinitely often, the PA estimator is

$$E[\frac{d^3W}{d\theta^3}]_{PA} = \lim_{M\to\infty} \frac{\sum_{m=1}^{M}\sum_{i=1}^{n_m}\sum_{j=1}^{i}\frac{d^3S_{jm}}{d\theta^3}}{\sum_{m=1}^{M}} \tag{13}$$

$$+ \lim_{M\to\infty}\frac{1}{M}\sum_{m=1}^{M}\left(3\lambda(\frac{d^2W_{n_m}}{d\theta^2}+\frac{d^2S_{n-1}}{d\theta^2})(\frac{dW_{n_m}}{d\theta}+\frac{dS_{n_m}}{d\theta})\right) \tag{14}$$

where M is the number of busy periods.

In order to show strong consistency, we must show

$$E[\frac{d^3W}{d\theta^3}]_{PA} = \frac{d^3E[W]}{d\theta^3}. \tag{15}$$

In general, both sides of (15) are intractable, but specializing to Poisson arrivals allows us to use the P-K formula from standard queueing theory for the right-hand side, and with further specialization to location or scale parameters yielding considerable simplification in (14). Before giving our main results we need the following lemmas.

LEMMA 1. *If θ is a location parameter for the random variable*

$$S = \theta + \phi(\omega) \quad \forall\omega\in\Omega, \quad \textit{for some function } \phi. \tag{16}$$

then

(i)

$$\frac{dS}{d\theta} = 1 \qquad \textit{and} \qquad \frac{d^2S}{d\theta^2} = 0,$$

(ii)

$$\frac{dE[S]}{d\theta} = E[\frac{dS}{d\theta}] = 1,$$

(iii)

$$\frac{dE[S^2]}{d\theta} = 2E[S\frac{dS}{d\theta}] = 2E[S].$$

LEMMA 2. *If θ is a scale parameter for the random variable*

$$S = \theta\phi(\omega) \quad \forall\omega\in\Omega, \quad \textit{for some function } \phi. \tag{17}$$

then

(i)

$$\frac{dS}{d\theta} = \frac{S}{\theta} \qquad \textit{and} \qquad \frac{d^2S}{d\theta^2} = 0,$$

(ii)

$$\frac{dE[S]}{d\theta} = E[\frac{dS}{d\theta}] = \frac{E[S]}{\theta},$$

(iii)

$$\frac{dE[S^2]}{d\theta} = 2E[S\frac{dS}{d\theta}] = \frac{2E[S^2]}{\theta}.$$

PROOF. See [Fu 1989]

THEOREM. *For the M/G/1 queue, if*
(a) θ is a location parameter of the service time distribution, or
(b) θ is a scale parameter of the service time distribution,
s.t. $E[S], E[S^2]$, and $dS/d\theta$ exist, where S is the service time random variable, then the perturbation analysis estimator of $d^3E[W]/d\theta^3$ (whose limit is given by below) is strongly consistent.

PROOF.

a) For θ a location parameter, the Lemma 1 gives $d^2S/d\theta^2 = 0$ and $dS/d\theta = 1$. Therefore, $d^3S/d\theta^3 = 0$. Substituting these simplifications into (14), we have

$$E[\frac{d^3W}{d\theta^3}]_{PA} = \lim_{M\to\infty} \frac{1}{M} \sum_{m=1}^{M} 3\lambda^2 \left(\sum_{i=0}^{\eta_{m-1}} \frac{dS_i}{d\theta}\right)^2 \left(\sum_{i=1}^{\eta_m} \frac{dS_i}{d\theta}\right) \tag{18}$$

$$= \lim_{M\to\infty} \frac{1}{M} \sum_{m=1}^{M} 3\lambda^2 \left(\sum_{i=1}^{\eta_{(m-1)}}\right)^2 (\sum_{i=1}^{\eta_m}) \tag{19}$$

$$= 3\lambda^2 \lim_{M\to\infty} \frac{1}{M} \sum_{m=1}^{M} \eta_{m-1}^2 \eta_m \tag{20}$$

where η_m is the number of customers served in the m-th busy period.

Since for the M/G/1 queue η_m and η_{m-1} are independent, we have

$$E[\frac{d^3W}{d\theta^3}]_{PA} = 3\lambda^2 E[\eta^2]E[\eta] \tag{21}$$

From standard queueing theory (e.g., [4]),

$$E[\eta^2] = \frac{1}{1-\rho} + \frac{2\rho}{(1-\rho)^2} + \frac{\lambda^2 E[X^2]}{(1-\rho)^3}. \tag{22}$$

and

$$E[\eta] = \frac{1}{(1-\rho)} \tag{23}$$

Thus

$$E[\frac{d^3W}{d\theta^3}]_{PA} = 3\lambda^2 \left(\frac{1}{(1-\rho)^2} + \frac{2\rho}{(1-\rho)^3} + \frac{\lambda^2 E[X^2]}{(1-\rho)^4}\right). \tag{24}$$

On the other hand, for the M/G/1 queue, we have the P-K formula (see, e.g., Kleinrock [4]):

$$E[W] = \frac{\lambda E[S^2]}{2(1-\lambda E[S])}. \tag{25}$$

Differentiating (25) w.r.t. θ three times and using Lemma 1 (ii) and (iii), we have

$$\frac{d^3E[W]}{d\theta^3}] = 3\lambda^2\left(\frac{1}{(1-\rho)^2} + \frac{2\rho}{(1-\rho)^3} + \frac{\lambda^2 E[X^2]}{(1-\rho)^4}\right). \tag{26}$$

Thus, (15) is satisfied.

b) Now assume that θ is a scale parameter, from Lemma 2, $dS/d\theta = S/\theta$ and $d^2S/d\theta^2 = 0$. We also have

$$\sum_{i=1}^{\eta_m} \frac{dS_i}{d\theta} = \frac{B_m}{\theta} \tag{27}$$

$$\frac{d^2W_{\eta_m}}{d\theta^2} = \lambda\left(\sum_{i=1}^{\eta_{(m-1)}} \frac{S_i}{\theta}\right)^2 = \frac{\lambda}{\theta^2}B_{m-1}^2 \tag{28}$$

where B_m is the length of the $m-th$ busy period.

Substituting these simplifications into (14), we have

$$E[\frac{d^3W}{d\theta^3}]_{PA} = \lim_{M\to\infty}\frac{1}{M}\sum_{m=1}^{M} 3\lambda^2\left(\sum_{i=0}^{\eta_{m-1}}\frac{dS_i}{d\theta}\right)^2\left(\sum_{i=1}^{\eta_m}\frac{dS_i}{d\theta}\right) \tag{29}$$

$$= \lim_{M\to\infty}\frac{1}{M}\sum_{m=1}^{M} 3\lambda^2\left(\sum_{i=1}^{\eta_{(m-1)}} B_{m-1}/\theta\right)^2 (\sum_{i=1}^{\eta_m})B_m\Phi/\theta \tag{30}$$

$$= 3\frac{\lambda^2}{\theta^3}\lim_{M\to\infty}\frac{1}{M}\sum_{m=1}^{M} B_{m-1}^2 B_m \tag{31}$$

Since B_m is i.i.d, B_m and B_{m-1} are independent. Under the assumption that the system regenerates infinitely often, taking $M \to \infty$, we have

$$E[\frac{d^3W}{d\theta^3}]_{PA} = 3\frac{\lambda^2}{\theta^3}E[B_{m-1}^2 B_m] = 3\frac{\lambda^2}{\theta^3}E[B^2]E[B], \tag{32}$$

From queueing theory

$$E[B^2] = \frac{E[S^2]}{(1-\rho)^3},$$

and

$$E[B] = \frac{E[S]}{(1-\rho)}$$

Thus,

$$E[\frac{d^3W}{d\theta^3}]_{PA} = 3\frac{\lambda^2}{\theta^3}\frac{E[S^2]}{(1-\rho)^3}\frac{E[S]}{(1-\rho)} = 3\frac{\lambda^2}{\theta^2}\frac{E[S^2]}{(1-\rho)^4} \tag{33}$$

On the other hand, for the M/G/1 queue, we have the P-K formula (see, e.g., Kleinrock [4]):

$$E[W] = \frac{\lambda E[S^2]}{2(1-\lambda E[S])}. \tag{34}$$

Differentiating (34) w.r.t. θ three times and using Lemma 2 (ii) and (iii), we get

$$\frac{dE[W^3]}{d\theta^3} = 3\frac{\lambda^2 E[S^2]}{\theta^2(1-\rho)^4} = 3\frac{\lambda^2 E[S^2]E[S]}{\theta^3(1-\rho)^4}$$

Thus, (15) is satisfied.

In general, we give any high order derivative estimator for the M/G/1 queue with location and scale parameters of service time distribution.

$$\frac{d^k W_n}{d\theta^k} = \begin{cases} \frac{d^k W_{n-1}}{d\theta^k} & \text{if } W_{n-1} + S_{n-1} \geq A_n; \\ k\lambda \frac{d^{k-1} W_{n-1}}{d\theta^{k-1}} (\frac{dW_{n-1}}{d\theta} + \frac{dS_{n-1}}{d\theta}) & \text{if } W_{n-1} + S_{n-1} < A_n, \end{cases} . \tag{35}$$

It is not difficutl to prove that for a location parameter

$$E[\frac{d^k W}{d\theta^k}]_{PA} = \frac{k!}{2}\lambda^{k-1} E[\eta^2] (E[\eta])^{k-2} \tag{36}$$

and for a scale parameter

$$E[\frac{d^k W}{d\theta^k}]_{PA} = \frac{k!}{2}\frac{\lambda^{k-1}}{\theta^k} E[B^2] (E[B])^{k-2} \tag{37}$$

The strongly consistency of PA estimators (36) and (37) can be verified by induction.

As we can see, tractability of (14) results from the property of the location or scale parameter given in Lemma 1 (i) or Lemma 2 (i), respectively, which allows us to convert the sum in (14) to the second moment of the number in a busy period or the second moment of the length of a busy period, respectively, for which analytic results exist for the M/G/1 queue.

Furthermore, the proofs expose the following interesting relationships relating any high order derivative of mean waiting time to second moments of quantities of the busy period distribution:

(a)

$$\frac{d^k E[W]}{d\theta^k} = \frac{k!}{2}\lambda^{k-1} E[\eta^2] (E[\eta])^{k-2},$$

where η is the number served in a busy period and θ is a location parameter.

(b)

$$\frac{d^k E[W]}{d\theta^k} = \frac{k!}{2}\frac{\lambda^{k-1}}{\theta^k} E[B^2] (E[B])^{k-2},$$

where B is the length of a busy period and θ is a scale parameter.

References

1.) Cassandras, C.G. "Discrete Event Systems: Modeling and Performance Analysis," Irwin, 1993.

2.) Fu, M. C. *On the Consistency of Second Derivative Perturbation Analysis Estimators for M/G/1 Queue*, Applied Mathematics Letters, 1989.

3). Glasserman, P. "Gradient Estimation Techniques for Discrete Event Systems," Kluwer, 1991.

4.) Ho, Y.C. and Cao, X.R., *Discrete Event Synamic Systems and Perturbation Analysis,* Kluwer Academic, 1991.

5.) Kleinrock, L., "Queueing Systems, Volume I," J. Wiley, 1975.

6.) Shi, L. *Discontinuous Perturbation Analysis of Discrete Event Dynamic Systems,* submitted to IEEE Trans. on AC., 1994.

7). Suri, R. and Zazanis, M., *Perturbation Analysis Gives Strongly Consistent Sensitivity Estimates for the M/G/1 Queue,* Management Science, **34**, pp.39-64, 1988.

8). Zazanis, M. and Suri, R., *Estimating Second Derivatives of Performance Measures of G/G/1 Queues Using a Single Sample Path,* 1989, submitted to Mathematics and Computers in Simulation.

Push and Pull Systems With External Demands

Michael Zazanis
Dept. of IEOR
University of Massachusetts
Amherst, MA 01003

September 1994

Abstract

We examine Push and Pull production control systems under a make-to-order policy with safety stock. We compute the distribution of the waiting time until a demand is satisfied, as well as flow time distributions and service levels for both systems. Work-in-process levels are determined as well. The analysis is carried out under Markovian assumptions and the explicit results on flow times depend on non-overtake conditions that are satisfied for single machine stations.

1 Introduction

This paper is a first step in an attempt to model the performance of various Push and Pull production strategies and the way they respond to external demands. These demands could represent either outside orders or signals from downstream cells in the same plant. The production line is modelled as a series of single machine stations with exponential processing times, external demands are Poisson, and, in this paper, two production control schemes are examined: A Push scheme with safety stock S and a Pull scheme with limited Work-in-process and safety stock S.

In the Push scheme each time a demand arrives it immediately authorizes the release of a new job. The demand is either immediately satisfied from the stock or it is backloged. The Pull scheme examined is similar to CONWIP (see [12, 11, 13]). The main difference is that instead of the requirement that the WIP in the system remain constant we require that the total work-in-process including the Finished Goods Inventory (FGI) remain constant and equal to S. External demands, if not immediately satisfied, are again backloged. In this scheme the arrival of a demand authorizes the release of a new job only when the work in process is less than S or equivalently the finished goods inventory is greater than zero. Proper operation of

this scheme, which ofcourse depends crucially upon the right choice for S, results in the same benefits in terms of increased system control as other pull systems (e.g. see [2, 6, 10, 9, 8, 12, 13]).

A number of performance criteria are considered, namely average WIP, the probability that a demand will be backloged, and the mean time to satisfy a demand. We also examine mean flow time as well as flow time variability and obtain explicit expressions in terms of the parameters of the system.

2 Systems under a Push Policy with Safety Stock

In this section, we examine in detail the push policy described above and obtain explicit expressions for its performance. The system consists of M single–machine stations in tandem, the processing times are exponential (with rates $\mu_i, i = 1, \ldots, M$), and the external demands are Poisson with rate λ. We will assume that the system is stable i.e. $\rho_i \stackrel{\text{def}}{=} \lambda/\mu_i < 1$ for all i. Finished jobs wait in the FGI buffer and, when a demand arrives, a new job will always be released to join the queue at the first station. Furthermore, if the FGI buffer is not empty, the demand is satisfied immediately and a finished job is removed from the system, otherwise the demand is backloged. We assume that initially there are S finished jobs in FGI (the safety stock) and no WIP. With this initial condition the work in process, (i.e. the unfinished jobs in the M stations) together with the finished jobs in the buffer is always greater than or equal to S.

Consider now the stationary version of this process. Restricting our attention to the M stations (and disregarding FGI), this system behaves as an open Jackson network. Let X_t^i be the WIP at machine i at time t and define

$$X_t = \sum_{i=1}^{M} X_t^i, \tag{1}$$

the total WIP in the system. Since the system is stationary,

$$P(X_0^1 = n_1, \cdots, X_0^M = n_M) = \prod_{i=1}^{M} (1 - \rho_i)\rho_i{}^{n_i}. \tag{2}$$

The negative part of $X_0 - S$, $Y_0 = (X_0 - S)^-$, is the finished goods inventory while the positive part, $Z_0 = (X_0 - S)^+$ is the number of backloged demands. Let τ denote the mean time to fill a demand in steady state. Applying Little's law to the finished goods inventory buffer, we obtain

$$\lambda\tau = E(X_0 - S)^+.$$

We proceed to compute τ explicitly in terms of the parameters of the system:

$$P(X_0 = k) = \sum_{\{\vec{n}: n_1 + \cdots + n_M = k\}} \prod_{i=1}^{M} \rho_i^{n_i}(1 - \rho_i) = G(M, k) \prod_{i=1}^{M} (1 - \rho_i),$$

with $G(M,k)$ given by:

$$G(M,k) \;=\; \sum_{\{\vec{n}\,:\,n_1+\cdots+n_M=k\}} \prod_{i=1}^{M} \rho_i^{n_i}.$$

Therefore,

$$\begin{aligned} E(X_t - S)^- &= \sum_{k=0}^{S} (S-k) P(X_0 = k) \\ &= \sum_{k=0}^{S} (S-k) G(M,k) \prod_{i=1}^{M} (1-\rho_i) \\ &= \prod_{i=1}^{M} (1-\rho_i) \sum_{k=0}^{S} (S-k) G(M,k)\,. \end{aligned}$$

Since $(X_0 - S)^+ = X_0 - S + (X_0 - S)^-$ and $EX_0 = \sum_{i=1}^{M} \frac{\rho_i}{1-\rho_i}$, we get

$$\tau \;=\; \frac{1}{\lambda} \left\{ \sum_{i=1}^{M} \frac{\rho_i}{1-\rho_i} \;-\; S \;+\; \prod_{i=1}^{M} (1-\rho_i) \sum_{k=0}^{S} (S-k) G(M,k) \right\}.$$

Anticipating (7) which is established in the next section,

$$\tau \;=\; \frac{1}{\lambda} \left\{ \sum_{i=1}^{M} \frac{\rho_i}{1-\rho_i} \;-\; S \;+\; \prod_{i=1}^{M} (1-\rho_i) \sum_{m=1}^{M} \frac{S - (S+1)\rho_m + \rho_m^{S+1}}{(1-\rho_m)^2 \prod_{l \neq m} (1 - \rho_l/\rho_m)} \right\}. \quad (3)$$

2.1 Customer service criteria

In this framework we can address a number of related issues pertaining to the level of service: The probability that a demand will be satisfied immediately is given by $P(S - X_0 > 0) = P(X_0 < S)$. In view of (1) and (2), the distribution of X_0 is the convolution of N independent geometric random variables (because of the independence of the number of customers in each station). To simplify the analysis we examine only the case where all stations have different utilizations. Then

$$P(X_0 = k) \;=\; \sum_{i=1}^{M} \rho_i^k \, \frac{\prod_{l=1}^{M} (1-\rho_l)}{\prod_{l \neq i} (1 - \rho_l/\rho_i)}. \quad (4)$$

The derivation of (4) is interesting since it does not make use of convolutions directly. The partial fractions expansion through which it is obtained is shown here in the case where $\rho_i \neq \rho_j$ for $i \neq j$. The z-transform of X_0 can then be written as

$$Ez^{X_0} \;=\; \prod_{i=1}^{M} \frac{1-\rho_i}{1-z\rho_i} \;=\; \sum_{i=1}^{M} \frac{A_i}{1-z\rho_i}$$

or

$$\prod_{i=1}^{M} (1-\rho_i) \;=\; \sum_{i=1}^{M} A_i \prod_{l \neq i} (1 - z\rho_l)\,.$$

Letting $z = \rho_i^{-1}$ gives

$$A_i = \frac{\prod_{l=1}^{M}(1-\rho_l)}{\prod_{l\neq i}(1-\rho_l/\rho_i)}, \qquad i = 1, 2, \ldots, M.$$

Hence,

$$E[z^{X_0}] = \sum_{i=1}^{M} \frac{1}{1-z\rho_i} \frac{\prod_{l=1}^{M}(1-\rho_l)}{\prod_{l\neq i}(1-\rho_l/\rho_i)}, \tag{5}$$

from which we obtain (4).

An alternative for computing the distribution of X_0 uses the normalization constants for CQNs for which a number of efficient computational algorithms exist.

$$\begin{aligned} P(X_0 = k) &= \sum_{\vec{n}:n_1+\cdots+n_M=k} \prod_{i=1}^{M}(1-\rho_i)\rho_i^{n_i} \\ &= G(M,k)\prod_{i=1}^{M}(1-\rho_i)\,. \end{aligned} \tag{6}$$

A comparison between (4) and (6) suggests the following expression for the normalization constant in a CQN

$$G(M,k) = \sum_{i=1}^{M} \frac{\rho_i^k}{\prod_{l\neq i}(1-\rho_l/\rho_i)}. \tag{7}$$

The above closed form expression for the normalization constant was first obtained in [7] (see also [5]).

2.2 Probability that a demand will be satisfied immediately

The probability that a demand will be satisfied immediately is given by

$$\begin{aligned} P(X_0 < S) &= \sum_{k=0}^{S-1} P(X_0 = k) \\ &= \prod_{i=1}^{M}(1-\rho_i) \sum_{k=0}^{S-1} G(M,k) \\ &= \sum_{i=1}^{M}\sum_{k=0}^{S-1} \rho_i^k \frac{\prod_{l=1}^{M}(1-\rho_l)}{\prod_{l\neq i}(1-\rho_l/\rho_i)} \\ &= \sum_{i=1}^{M}(1-\rho_i^S) \prod_{l\neq i} \frac{1-\rho_l}{1-\rho_l/\rho_i}. \end{aligned} \tag{8}$$

3 Limited WIP systems with safety stock

We now consider a make–to–order system with safety stock under a Limited WIP policy. As before the cell consists of M single–machine stations (exponential processing times with rates μ_i). External (or down stream) demands are Poisson with rate λ

and, when the system starts, there are S finished parts in Finished Goods Inventory (FGI) and no parts are being processed. Denote by X_t^i the number of parts in process or waiting in front of machine i at time t, and by $X_t = \sum_{i=1}^{M} X_t^i$ the total WIP. When a demand arrives, if there is a part in FGI then it is satisfied immediately and a new part is authorized to start processing at (or join the queue in front of) station 1. If however upon the arrival of the demand no parts are available in FGI then the demand is backloged until a part is finished. Backloged demands are satisfied on a FCFS basis. Furthermore, *a backloged demand does not authorize the production of a new part.* Let Y_t denote the number of parts in FGI with negative values of Y_t corresponding to the number of backloged orders. Under the above policy, $X_t \leq S$ with equality holding whenever $Y_t \leq 0$ (i.e. whenever there are no parts in FGI but, possibly, a demand backlog).

To analyze the performance of this policy we consider now a system, equivalent to the one just described under our Markovian assumptions. It consists of an *open* tandem network with M single–server exponential stations (with the same processing rates). This system has a global buffer of size S (i.e. the total number of customers in the network, X_t, is constrained to be $\leq S$. Customers arrive from outside according to a Poisson process with rate λ and if upon arrival the number of parts in the network is equal to S then they wait in front of the global buffer according to a FCFS discipline. As soon as a customer finishes processing at station M, a customer waiting outside the global buffer is immediately admitted and joins the queue, or starts processing, at station 1. Let Z_t be the total number of customers at time t, including those waiting outside the buffer. When $Z_t \leq S$ in this equivalent system, the original system has no backloged demands, Z_t parts in process (WIP) and $S - Z_t$ parts in FGI. When $Z_t > S$, there are S parts in process (the maximum allowed), $Z_t - S$ unsatisfied demands, and of course no parts in FGI.

If the probability that a demand is backloged is small the operation of this system can be adequately approximated by a closed queueing network with the FGI buffer acting as a single server station with rate λ. (For details see [14].) In the next station we examine flow times in such systems.

4 Flow time distributions for closed queueing networks

Flow times in closed queueing networks were studied in Boxma, Kelly, and Könheim (1984), and Daduna (1982). The pertinent results can be summarized as follows: If T_i is the flow time through the i'th station, the joint Laplace transform for the flow times of a tagged customer through the stations satisfies the following product form relationship

$$E[e^{-s_1 T_1 - \cdots - s_M T_M}] = \sum_{\vec{j} \in \mathcal{S}(M, N-1)} p(j_1, \ldots, j_M) \prod_{i=1}^{M} \left(\frac{\mu_i}{\mu_i + s_i} \right)^{j_i + 1}, \qquad (9)$$

where $\mathcal{S}(M,N) = \{\vec{j} : j_1 + \cdots + j_M = N\}$. (The above is true provided that a non-overtake condition holds, which of course is the case for cyclic, single server networks.) Setting $\rho_i = 1/\mu_i$, the rhs of the above equation can be written as

$$\sum_{\vec{j}\in\mathcal{S}(M,N-1)} \frac{\rho_1^{j_1}\cdots\rho_M^{j_M}}{G(M,N-1)} \prod_{i=1}^{M} \left(\frac{\mu_i}{\mu_i + s_i}\right)^{j_i+1},$$

or, equivalently,

$$\prod_{i=1}^{M} \left(\frac{\mu_i}{\mu_i + s_i}\right) \sum_{\vec{j}\in\mathcal{S}(M,N-1)} \prod_{i=1}^{M} \left(\frac{1}{\mu_i + s_i}\right)^{j_i} \frac{1}{G(M,N-1)}. \tag{10}$$

We now use these results in conjunction with (7) to obtain an explicit formula for the flow time distribution. The first factor in the above expression corresponds to the joint Laplace transform of the processing times for a job while the second to the joint Laplace transform of waiting times at the stations. Setting $s_1 = \cdots = s_M = s$ in (10) and taking into account (7) we obtain the following expression for the Laplace transform of the cycle time:

$$Ee^{-sT} = \frac{\sum_{m=1}^{M} \frac{\alpha_m}{(\mu_m + s)^N}}{\sum_{m=1}^{M} \frac{\alpha_m}{\mu_m^N}}, \tag{11}$$

where

$$\alpha_m = \prod_{l\neq m} \frac{1}{\mu_l - \mu_m}. \tag{12}$$

From (11) we readily obtain the following expressions for the moments of the cycle time:

$$ET^k = N(N+1)\cdots(N+k-1) \frac{\sum_{m=1}^{M} \alpha_m \mu_m^{-(N+k)}}{\sum_{m=1}^{M} \alpha_m \mu_m^{-N}}, \quad k = 1,2,\ldots. \tag{13}$$

The throughput of the closed system with N customers can easily be computed from the first moment of the cycle time (13) and Little's Law:

$$\lambda(N) = \frac{\sum_{m=1}^{M} \alpha_m \mu_m^{-N}}{\sum_{m=1}^{M} \alpha_m \mu_m^{-(N+1)}}. \tag{14}$$

From (13), (14) we also obtain the following version of Little's Law for the moments of the cycle time in a closed network:

$$\lambda(N)\lambda(N+1)\cdots\lambda(N+k-1)\,ET^k \;=\; N(N+1)\cdots(N+k-1)\,. \qquad (15)$$

The coefficient of variation of the cycle time with N customers in the system can be written as

$$C_v(N) \;=\; \sqrt{\left(1+\frac{1}{N}\right)\frac{\lambda(N)}{\lambda(N+1)}-1}\,. \qquad (16)$$

Given that $\lambda(N)$ is increasing in N we have the (asymptoticaly tight) bound:

$$C_v(N) \leq \sqrt{\frac{1}{N}}\,.$$

References

[1] Boxma, O.J., F.P. Kelly, and A.G. Konheim (1984). "The Product Form for Sojourn Time Distributions in Cyclic Exponential Queues," *J. ACM,* **31,** 1, 128-133.

[2] Buzacott, J.A. and J.G. Shanthikumar (1993). *Stochastic Models of Manufacturing Systems*, Prentice Hall, 490-515.

[3] Daduna, H., (1982). "Passage times for overtake-free paths in Gordon-Newell networks," *Adv. Appl. Probab.* **14,** 672-686.

[4] Delersnyder, J.-L., T.J. Dodgson, H. Muller, and P.J. O'Grady (1989). "Kanban controlled pull systems: an analytic approach", *Management Science,* **35,** 1079-1091.

[5] Gordon, J.J. (1990)."The Evaluation of Normalizing Constants in Closed Queueing Networks," *Opns. Res.* **38,** 5, 863-869.

[6] Hall, R. W. (1983). *Zero Inventories*, Dow Jones-Irwin, Homewood, Ill.

[7] Harrison, P.G. (1985). "On Normalizing Constants in Queueing Networks," *Opns. Res.* **33,** 464-468.

[8] Karmarkar, U. S. (1986). "Push, Pull, and Hybrid Control Schemes", Working Paper Series No. QM8614, Graduate School of Management, University of Rochester, Rochester, New York.

[9] Mitra, D., and I. Mitrani (1990). "Analysis of a Kanban discipline for cell coordination in production lines, I", *Management Science,* **36,** 1548-1566.

[10] Mitra, D., and I. Mitrani (1991). "Analysis of a Kanban Discipline for Cell Coordination in Production Lines, II: Stochastic Demands", *Operations research*, Vol. 39, No. 5, 807-823.

[11] Spearman, M. (1992). "Customer Services in Pull Production Systems", *Operations research*, Vol. 40, No. 5, 948-958.

[12] Spearman, M., D. Woodruff, and W. Hopp (1990). "CONWIP: a Pull Alternative to Kanban ", *Int. j. Prod. Res.*, **28**, 5, 879-894.

[13] Spearman, M., and M. Zazanis (1992). "Push and Pull Production Systems: Issues and Comparisons", *Operations research*, Vol. 40, No. 3, 521-531.

[14] Zazanis, M.A. (1994). "Pull Systems with Limited WIP", Technical Report, IEOR Dept. Univ. of Massachusetts, Amherst, MA 01003.

THE SUPERVISORY CONTROL OF AN EXPERIMENTAL MANUFACTURING CELL

Bertil A. Brandin
Department of Electrical and Computer Engineering
University of Toronto, 10 King's College Road
Toronto, Ontario M5S 1A4, Canada
brandin@control.utoronto.ca

keywords: supervisory control, discrete event systems, automated manufacturing systems

ABSTRACT

A supervisory control system, based on controlled automata concepts, is developed for an experimental manufacturing cell[1]. The approach adopted guarantees that: (i) the resulting controlled behaviours do not contradict the behavioural specifications and are nonblocking, (ii) the controlled behaviours are maximally permissive within the behavioural specifications. Both centralised and modular supervision are considered. The automata models for the complete cell are provided together with the corresponding supervisors. The supervisory control system implementation presented only requires the use of `off-the-shelf' industrial control systems.

1 INTRODUCTION

Automated manufacturing systems typically incorporate the following: a number of numerically controlled part processing machines, material handling devices, part inspection/testing devices, in-process part storage systems, and a supervisory control system [2]. The latter performs the following three tasks: supervisory control, communication and housekeeping. Supervisory control consists of (i) the monitoring of the workcell behaviour via sensory feedback, (ii) control evaluation in accordance with a supervisory control law which maps the workcell behaviour to corresponding controls, and (iii) control enforcement via the downloading and execution of the appropriate device programs. Communication allows sensory feedback and control enforcement to be performed. Housekeeping is the set of tasks related to supervisory control and communication which are necessary to their implementation. Supervisory control system development consists in the procedure generating supervisors and control laws which specify how the supervisory control system is to react to the manufacturing system behaviour in order to satisfy behavioural specifications (routing, sequencing, safety,

error recovery...) [2]. To develop supervisory control systems, different techniques such as knowledge engineering [7], Petri nets [8] and controlled automata may be exploited [1, 3, 4, 5, 6, 10, 11]. The work presented is based on the latter, since these provide important advantages over other approaches, as is described in Section 2.

---

[1] Computer Integrated Manufacturing Laboratory (CIMLab), Department of Mechanical Engineering, University of Toronto.

2 THE SUPERVISORY CONTROL OF DES

Discrete event systems (DES) are dynamic systems that evolve in accordance with the abrupt occurrence of events [10]. They are generally *asynchronous* (not clock driven) and *non deterministic* (some events may occur spontaneously). Such systems are encountered in a variety of fields, for example in manufacturing, robotics, computer and communication networks, traffic, and logistics. The supervisory control of DES in accordance with behavioural specification is a new research area which is receiving increasing recognition. The approach is based on information feedback on the occurrence of events and controlled automata concepts. It offers important advantages over other approaches [2]:

- controlled behaviours are guaranteed not to contradict the behavioural specifications and to be nonblocking;
- controlled behaviours are optimal: maximally permissive within the behavioural specifications; and
- modular supervisors are guaranteed to be nonconflicting.

The development of a supervisory control system is divided into three main steps [2]:

(i) The modelling of the plant and behavioural specifications to be enforced: Both the plant physical behaviour and the behavioural specifications are translated in the form of automata. Furthermore, events are subdivided into *controllable* events, i.e. the events which may be disabled through control, and *uncontrollable* events, i.e. the events which may not be disabled through control [10].

(ii)

adopted, the automata representing the physical behaviour of the system to be controlled and the automata representing the corresponding specifications are fed to a computer program which will (i) tell us whether it is possible for the system to behave within the specifications, and (ii) return the supervisor(s) embodying the maximally permissive controlled behaviour of the system within the latter specifications, and the corresponding control law(s) yielding such behaviour[2]. Centralised and modular supervisory control are briefly discussed below. For an extensive treatment, the reader is referred to [5,10,11]. Centralised supervision implies that the overall supervisory task is carried out by one single supervisor embodying the closed-loop behaviour of the plant. Furthermore such a supervisor must be provided with complete information on the occurrences of events in the plant. Modular supervision implies the division of the overall supervisory task into two or more subtasks, and the resulting individual subsupervisors are run concurrently to implement a solution of the original control problem [11]. A modular supervisor is ideally more readily modified, updated and maintained than its centralised counterpart. For example, if one subtask is changed, then it should only be necessary to redesign the corresponding component supervisor. Unfortunately, these advantages are not always to be gained without a price [5,11]. The fact that the individual supervisory modules are simpler implies that their control action must be based on an aggregated version of the global system state [5,11]. Consequently, the different component supervisors, acting quasi-independently on the basis of partial information, may come into conflict and the overall system may fail to be nonblocking. Thus a fundamental issue that always arises in

[2]Let the automaton S be a supervisor for a plant G. Let x be a state of S. The corresponding supervisory control law V maps every state x of S to the a set of legal transitions V(x) [2, 5, 10].

the presence of modularity is how to guarantee the nonconflictuality of the various supervisors working together. Shared resources are a good example of a possible source of conflict between supervisors [5]. Centralised supervisory control is seen to be suitable for fairly small problems (in the order of 10^4 states, for plant and specifications combined) since the corresponding supervisory controls tend to be large in state size; they are required to encompass the behaviour of the whole plant subject to all specifications. Modular supervisory controls jointly achieve equivalent system behaviours but tend to be smaller in state size, even though they are generally computationally more expensive. However, corresponding off-line computations may often be carried out in parallel, thus reducing the effective required computation time [2].

(iii)PLC based implementation: The supervisor(s) and the corresponding control law(s) obtained are coded in a programmable logic controller which accordingly reacts to event occurrences. Each PLC carries out: • the monitoring of the plant behaviour via sensory feedback, typically using polling: during every PLC scan [2,9], the status of the various inputs is checked; • control evaluation in accordance with supervisor(s) and corresponding control law(s), and • control enforcement, which is carried out by the programmable logic controller output modules upon the energising of given outputs [2,9].

3 Experimental Setup

3.1 Problem Description

The automated manufacturing cell considered consists of a robot, a vision system, a conveyor belt, two simulated numerically controlled machines, a mill and a drill, and a supervisory control system constituted by Programmable (Logic) Controllers. It is shown in Figure 1. Two types of workpieces must be worked on by both the mill and the drill before being placed in the bin. Workpieces arrive in random order on the conveyor belt. Supervisory control may be carried out in either centralised or modular fashion.

3.2 Modelling

3.2.1 Robot and Vision System

A GMF S-100 robot with a Karel R-F controller, is used to transfer workpieces within the cell. Along with its grasping tool it is equipped with a camera for workpiece recognition, linked to the workcell vision system. The robot is programmed to carry out seven tasks. These are:

- go to conveyor and grasp, and go to mill and drop,
- go to drill and grasp, and go to mill and drop,
- go to mill and grasp, and go to drill and drop,
- go to conveyor and grasp, and go to drill and drop,
- go to mill and grasp, and go to bin and drop,
- go to drill and grasp, and go to bin and drop, and
- recognition.

The robot is hardwired to the P(L)C with four output internals, which are energised in patterns in order to activate corresponding robot tasks.
The vision system is characterised by object (workpiece) premarking, and active visual perception. Objects are modelled by a small number of their two-dimensional perspective views, referred to as *standard views*, each associated with a corresponding *standard view axis*. The mobile camera attached to the robot arm is used to acquire a standard view of the object (workpiece) to be recognised. Local surface normals are defined for each standard view by premarking the object with red circular markers (defining corresponding standard view axis). Initially the vision system attempts to detect a marker in the field of view according to a pre-planned search strategy. In case a marker is detected, the corresponding standard view is estimated, and the surface normal and marker centre are computed with respect to the camera frame. Subsequently, through the alignment of the camera's optical axis with the estimated surface normal, a standard view of the object is acquired. Depending on the degree of similarity between the acquired standard view and the reference standard views of all the possible objects considered, one or more candidates are selected. In case only one matching reference standard view is found, successful recognition is assumed. In case several matching reference standard views are found, one additional standard view is acquired. In case no unique candidate or no candidate is found, the object is assumed to be unrecognisable. Finally, having recognised an object, its orientation and position are determined. In this example, recognition will always be assumed to be successful, and the vision system will be able to distinguish between two types of workpieces: type 1 and type 2. The robot and vision system are modelled as shown in Figure 2. A complete list of events for the experimental cell is provided in Table 2.

3.2.2 The Conveyor positioning system

The conveyor used is an Amatrol 83-129 loop conveyor with a 14130 pallet positioning system. Its purpose is the transportation of incoming workpieces. The pallet positioning system consists of:

- a positioner limit switch which is attached on the inner conveyor rail, just upstream of the pallet positioner, it is placed so that the incoming pallet will trip it, thus signalling the pallet's presence;
- two gage stops which are mechanical stops mounted on either side of the conveyor belt immediately downstream from pallet positioner;
- a restraint pin located upstream from the limit switch along the inside rail of the conveyor, its purpose is to queue pallets for the pallet positioning system;
- a pallet positioner which consists of four lifting rods which straddle the belt in a rectangular formation, their purpose is lifting the pallet above the moving conveyor belt; and
- two timers used two insure that the pallets reach the desired position before acting on either the restraint pin or the pallet positioner (see corresponding specifications); the timers are actually programmed in the supervisory control system.

The various actuators of the pallet positioning system are powered by compressed air and controlled with solenoids. The conveyor positioning system is modelled as shown in Figure 3.

3.2.3 The Simulated Numerically Controlled Machines

The numerically controlled machines considered are simulated. They are composed of platform on which workpieces are positioned by the robot and of mechanical switches which indicate their presence or otherwise. The switches are also hardwired to the P(L)C. Worktime is simulated by time delays programmed on the P(L)C. They are modelled as shown in Figure 4. Since for control purposes, only the end of operations by the mill and drill are of interest, these models are simplified and are shown in Figure 5.

3.2.4 The Supervisory Control System

The supervisory control system is composed of two programmable logic controllers Allen-Bradley PLC-5 linked by a Allen-Bradley Data Highway Plus network to a personal computer used for programming purposes.

3.2.5 Behavioural Specifications

Four behavioural specifications must be enforced through supervisory control. These are:

- Two types of workpieces arrive on pallets in the workcell in random order. Type 1 workpieces must be worked on first by the mill, then by the drill and subsequently evacuated to the bin. Type 2 workpieces must be worked on first by the drill, then by the mill and subsequently evacuated to the bin. The automata shown in Figure 6 enforce the above specification. Type 1 workpieces are worked on first by the mill (op11, end1), then by the drill (op21, end2) and are subsequently evacuated to the bin (exi2). Type 2 workpieces are worked on first by the drill (op22, end2), then by the mill (op12, end1) and are subsequently evacuated to the bin (exi1). This specification is implicit in the automata S4 to S7 (cf. Figure 6): either machine may not start operating on a new workpiece before the workpiece currently on the machine has been transferred onwards.
- The various actuators of the conveyor belt positioning system must be coordinated so that only one pallet at time is halted and lifted from the conveyor, the remaining pallets being queued upstream.
- The automaton shown in Figure 7 enforces the above specification. Initially the arrival of a pallet triggers the positioner limit switch (sens). Subsequently, both the gage stops and the restraint pin are set in the up position (gaup, piup), thus blocking the pallet that has arrived, and queuing arriving pallets. Furthermore, before setting the pallet positioner in the up position (poup), a time delay of two seconds (wai2) is introduced allowing the pallet to travel to the gage stop. The recognition and transfer of the workpiece may now take place: the pallet will not move until a pallet advance (adpa) is requested. Then, the gage stops and the pallet positioner are set in the down position (gado, podo). Before setting the restraint pin in the down position (pido), to allow a new pallet arrival, a time delay of four seconds (wai4) is introduced letting the pallet present in the positioning system to travel downstream of the gage stop. The events (wai2) and (wai4) are assumed internal to the supervisory control system.
- Workpiece recognition may only be carried out upon arrival of a pallet at the conveyor belt presentation system. The automaton shown in Figure 8 enforces the above specification. The recognition procedure must be intialised (inre) and either a workpiece of type 1 or of type 2 must be recognised before the robot is allowed to transfer it. Upon transfer of the workpiece, the pallet may be advanced (adpa), allowing the arrival of a new workpiece.

3.3 Supervisors and Control Laws Synthesis

The cell, constituted by the two machines and the pallet positioning system, is represented by an automaton with 256 states and 2816 transitions. The specifications grouped together are represented by an automaton with 540 states and 1977 transitions. With centralised supervision [10], the corresponding supervisor embodying the controlled behaviour of the plant within the specifications is an automaton with 540 states and 1329 transitions. The latter is not shown due to space limitations.

With modular supervision under partial observations [5], it is possible to supervise the manufacturing cell, and obtain the same controlled behaviour obtained with centralised supervision, with six supervisors functioning concurrently, namely the specification automata S4 to S9. This approach requires the coding in ladder logic of 29 states (3 states each for specifications S4 to S7, 12 states for specification S8, and 5 states for specification S9) compared to 540 states when considering centralised supervision.

3.4 Supervisors and Control Laws Implementation

The supervisors S4, S5, S6, S7, and S9, responsible for the behaviour of the robot and the vision system, and the numerically controlled machines, were coded on one programmable controller (PLC0) and supervisors S8, responsible for the behaviour of the pallet positioning system, was programmed on a second programmable controller (PLC1) (see Table 1 and Figure 1). The programmable controllers and corresponding supervisors implement the behavioural specifications concurrently and without conflict.

PLC0	S4, S5, S6, S7, S9
PLC1	S8

Table 1: PLCs and corresponding supervisors

6 Conclusions

The development of a supervisory control system for an experimental automated manufacturing system was presented. The work illustrated the viability of automata based approaches, in particular modular supervision under partial observation. Computational tools used were used to synthesise the supervisors and corresponding control laws, and/or to verify and guarantee the supervisors nonconflicting nature. The plant controlled behaviour obtained was observed never to violate the specifications considered, and was nonblocking as guaranteed by the theory. Finally, the implementation of the supervisory control system only required the use of `off-the-shelf' industrial programmable logic controllers.

The results of this work are currently being applied to the supervisory control of an automated assembly cell consisting of 3 robotic assembly stations linked by five conveyor belts. Preliminary results show that the supervisory control system development approach presented is applicable to such complex systems (approximately 400 inputs and outputs, with state representation in the order of 10^{10} states).

7 ACKNOWLEDGEMENTS

The author would like to thank Peter Chen for his help with the experimental setup, Technology Ontario and Allen-Bradley (Canada) for their support of the Ontario/Rhône-Alpes Collaborative Project on Automated Manufacturing.

8 BIBLIOGRAPHY

[1] S. Balemi, ``Discrete-event systems control of a rapid thermal multiprocessor", INCOM 92, Toronto, May 1992, pp. 53-58.

[2] B.A. Brandin, The Supervisory Control of a Small Manufacturing Cell, SCG Report 9404, Department of Electrical and Computer Engineering, University of Toronto, April 94.

[3] B. A. Brandin, W. M. Wonham and B. Benhabib, "Discrete event supervisory control applied to the management of manufacturing workcells", 7th international Conference on Computer Aided Manufacturing Engineering, V.C. Venkatesh and J.A. McGeough Eds., Elsevier 1991, pp. 527-536.

[4] B. A. Brandin, W. M. Wonham, "The supervisory control of timed discrete-event systems", IEEE Transactions on Automatic Control, Vol. 39, No. 2, February 1994, pp. 329-342

[5] B. A. Brandin, W. M. Wonham, "Modular supervisory control of timed discrete-event systems", 1993 IEEE Control and Decision Conference, San Antonio, December 1993, pp. 2230-2235.

[6] B. A. Brandin, W. M. Wonham and B. Benhabib, "Manufacturing cell supervisory control - a modular timed discrete-event system approach", 1993 IEEE International Conference on Robotics and Automation, Atlanta, May 1993, pp. 846-851.

[7] E. Coste-Manière, B. Espiau, and E. Rutten, ``A task level robot programming language and its reactive execution", 1992 IEEE International Conference on Robotics and Automation, Nice, May 1992, pp. 2751-2756.

[8] R. David, H. Alla, ``Autonomous and timed continous Petri nets", 11th International Conference on Application and Theory of Petri Nets, Paris, June 1990.

[9] G. Michel, ``Programmable Logic Controllers - Architecture and Application", Wiley 1990.

[10] P. J. Ramadge and W. M. Wonham, ``The control of discrete-event systems", IEEE Proceedings, Vol. 77, No. 1, January 1989, pp. 81-98.

[11] W. M. Wonham and P. J. Ramadge, ``Modular supervisory control of discrete event systems", Maths. of Control, Signals and Systems, Vol. 1, No. 1, 1988, pp. 13-30.

events (c: controllable, u: uncontrollable)		
inre	initialise recognition	c
rec1	recognition of a workpiece of type 1	u
rec2	recognition of a workpiece of type 2	u
op11	go to conveyor and grasp, go to mill and drop	c
op12	go to drill and grasp, and go to mill and drop	c
op21	go to mill and grasp, and go to drill and drop	c
op22	go to conveyor and grasp, go to drill and drop	c
exi1	go to mill and grasp, and go to bin and drop	c
exi2	go to drill and grasp, and go to bin and drop	c
sens	positioner limit switch senses a pallet	u
gaup	gage stops is placed in the up position, blocking a pallet in the positoner system	c
gado	gage stops is placed in the down position	c
piup	restraint pin is placed in the up position, queuing pallets upstream	c
pido	restraint pin is placed in the down position	c
poup	pallet positioner is placed in the up position, lifting pallet from the conveyor belt	c
podo	pallet positioner is placed in the down position	c
stw2	start time out of 2 seconds	c
stw4	start time out of 4 seconds	c
wai2	time out of 2 seconds	u
wai4	time out of 4 seconds	u

Table 2: Description of events

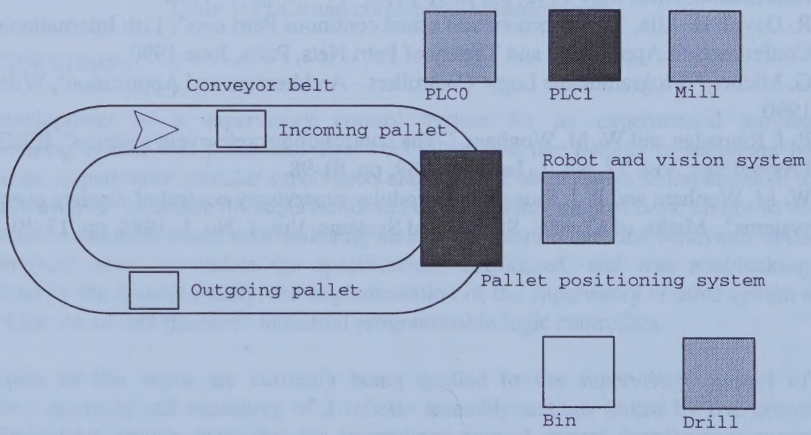


Figure 1: The experimental cell and PLCs (supervisory control system)

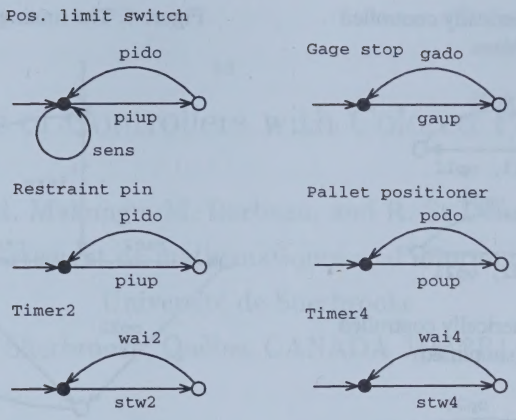


Figure 3: The conveyor positioning system

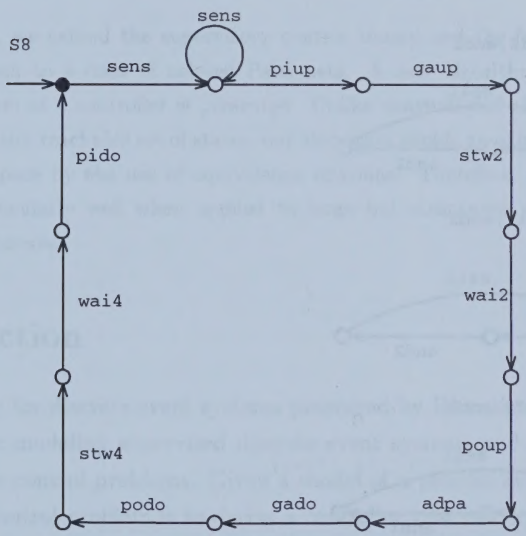


Figure 7: The positioning system specification

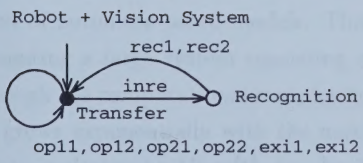


Figure 2: The robot and vision system

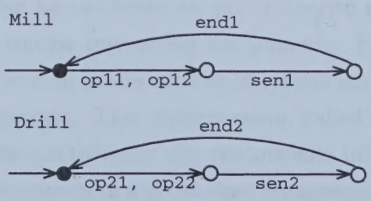


Figure 4: The numerically controlled machines

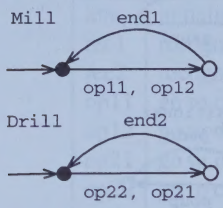


Figure 5: The numerically controlled machines (simplified)

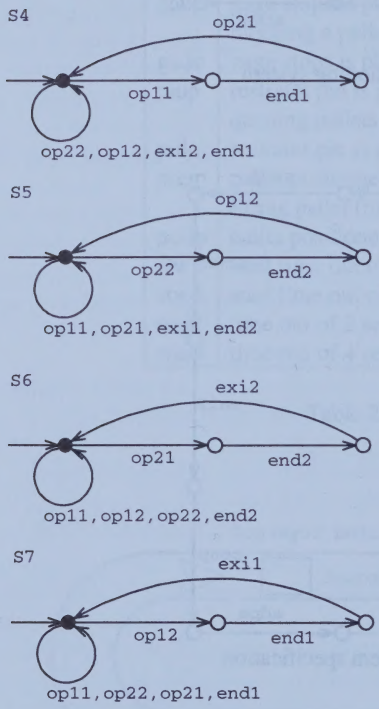


Figure 6: The routing specifications

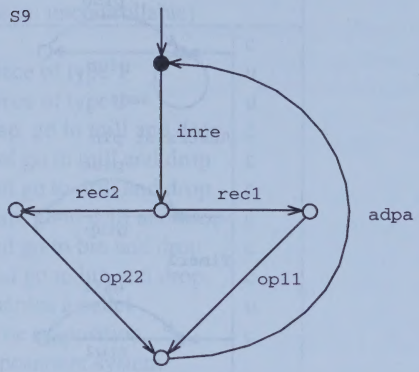


Figure 8: The pallet-advance specification

Synthesis of Controllers with Colored Petri Nets*

M. Makungu, M. Barbeau, and R. St-Denis
Département de mathématiques et d'informatique
Université de Sherbrooke
Sherbrooke, Québec CANADA J1K 2R1

Abstract

In this paper, we extend the supervisory control theory and the forbidden state control problem to a class of colored Petri nets. A new algorithm for deriving a compact form of a controller is presented. Unlike conventional algorithms that explore the entire reachable set of states, our algorithm avoids an exhaustive search of the state space by the use of equivalence relations. Therefore, our algorithm performs particularly well when applied to large but structured processes with similar components.

1 Introduction

The control theory for discrete event systems pioneered by Ramadge and Wonham [10] is a framework for modeling supervised discrete event systems and applying synthesis algorithms to solve control problems. Given a model of a process and a control specification, the basic control problem is to derive a controller that restricts the operation of the process such that the constraints included in the control specification are satisfied.

The supervisory discrete event control theory has been primarily studied in the context of automata-based models. This approach can be cumbersome, particulary in representing a large system consisting of numerous similar interacting components. Even though the computational complexity can be polynomial in the number of system states, it grows exponentially with the number of components. This phenomenom, called the *state explosion problem* [1], can be investigated by partitioning the components into a small number of equivalence classes so that all components in a given class are essentially

*The research described in this paper was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Fonds pour la formation de chercheurs et l'aide à la recherche (FCAR).

similar. This paper makes a contribution in this direction by designing an algorithm that reduces the complexity of the supervisory control problem under full observation when applied to large but structured processes with similar components.

Recently, Petri nets have begun to receive attention as alternative models for investigating the control of discrete event systems [9] [2] [4] [12] [3] [8] [11] [5]. Among the Petri net-based models, colored Petri nets are powerful enough to describe complex systems in a manageable way, particularly when the systems contain many interacting components that are similar but not identical. It is well-known that the class of colored Petri nets (CP-nets) with finite color sets is equivalent to the class of place/transition nets (PT-nets). The CP-nets describe the same systems as the PT-nets but often in a more compact form and allow the use of equivalence classes to represent similar components [7]. Specifications can be more readable for human beings and eventually more tractable.

The work presented in this paper addresses the forbidden state control problem for a class of CP-nets in which their color sets are finite and their behavior satisfies boundedness properties. From a control specification expressed as a set of forbidden markings, a computationally procedure determines the unique maximal set of admissible markings and derives a controller by means of a reachability tree [6] with equivalence relations.

The layout of the paper is as follows. The next section presents an overview of the supervisory control theory when the process is specified by a CP-net and formulates the control synthesis problem for avoiding a set of forbidden markings. Section 3 briefly describes the new synthesis algorithm and illustrates its application with an example. Finally, concluding remarks are included in Section 4.

2 The Supervisory Control Theory and the Forbidden State Control Problem

In this section, we assume that the reader is familiar with the notation of CP-nets introduced by Jensen. The reader is referred to his book [7] for a detailed exposition. As an example, Fig. 1 shows a CP-net that models a unidirectional circular railway with two trains [7]. The track is divided into six different sections. A section is represented by a place (a circle) and the passage from a section to the adjacent section is modeled by a transition (a rectangle). Places can be marked by tokens. In ordinary Petry nets, tokens are unstructured and undistinguishable. In CP-nets, tokens are, possibly structured, values. In our example, token values model identities of trains. The declaration "color $Train = Tr1 \mid Tr2$" defines the domain of token values. Every place may contain values from the domain $Train$. A given token marking a given place represents the fact that the corresponding train occupies the corresponding section. Initially, train number 1 ($Tr1$)

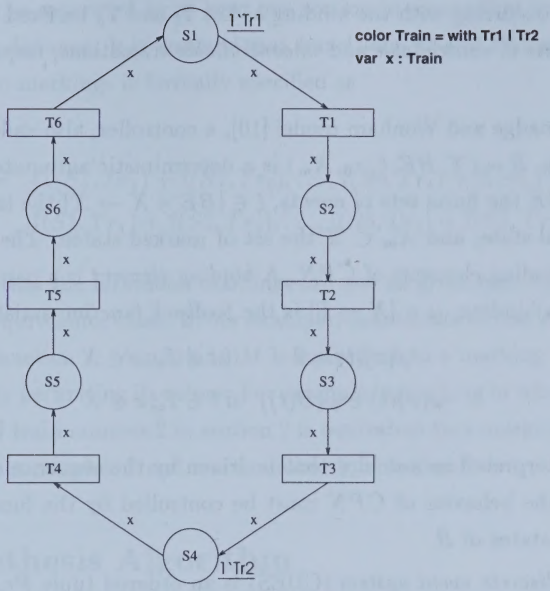


Figure 1: A CP-net describing a circular railway with two trains

is in section 1 and train number 2 ($Tr2$) in section 4. The initial marking is represented by the expressions $1`Tr1$ and $1`Tr2$ next to places $S1$ and $S4$ respectively. The integer before the back quote indicates the number of occurrences of the value after the back quote. The execution of a transition always extracts one token x from its incoming place and inserts one token x in its outgoing place. Since the variable x ranges over the color set $Train$, x may only take either value $Tr1$ or $Tr2$. In this model, no constraint is given concerning the position of the two trains relative to each other.

The basic problem in supervisory control is to construct a controller that can turn off various events of an uncontrolled discrete event system (DES) called a *process* (Pr). We consider a process modeled by a CP-net noted CPN. This problem is decidable if color sets and contents of places are bounded. Therefore, we assume that the color sets are finite and the process Pr is defined as a pair (CPN, K), where K is a *capacity function* associating a bound to the contents of every place. In our example, capacity is two.

A *binding* is a set of variable-value associations for the variables surroundings a transition. A binding is written in the form $< v_1 = c_1, \ldots v_n = c_n >$ where v_i is a variable and c_i a color. Let T be the set of transitions of CPN. Let $B = \bigcup_{t \in T} B(t)$, the set of all *bindings* for CPN, where $B(t)$ is the set of all bindings for a transition t. In our example, $B = \{< x = Tr1 >, < x = Tr2 >\}$. Let $\wp(B)$ denote the power set of B. Let $\Gamma = [T \rightarrow \wp(B)]$ be the set of all functions γ, called *control patterns*, that assign a subset of $B(t)$ to the every transition of T. If $b \in \gamma(t)$, then the controller prevents the

transition t from occurring with the binding b. Let T_c and T_u be fixed disjoint subsets of T denoting the sets of *controllable* and *uncontrollable* transitions, respectively, such that $T = T_c \cup T_u$.

As in the Ramadge and Wonham model [10], a controller, also called a supervisor, is a pair $S = (R, \varphi)$. $R = (X, BE, \xi, x_0, X_m)$ is a deterministic automaton where X is the finite state set, BE the finite sets of events, $\xi \in [BE \times X \to X]$ the transition function, $x_0 \in X$ the initial state, and $X_m \subset X$ the set of marked states. The set of events BE is the set of all binding elements of CPN. A *binding element* is a pair (t, b) where t is a transition and b a binding. $\varphi \in [X \to \Gamma]$ is the *feedback function* satisfying:

$$\begin{aligned} \varphi(x)(t) &= \emptyset \qquad \text{if } t \in T_u, x \in X \\ \varphi(x)(t) &\in \wp(B(t)) \quad \text{if } t \in T_c, x \in X \end{aligned}$$

R must be interpreted as a device that is driven by the sequence of steps generated by CPN, while the behavior of CPN must be controlled by the function φ, and thus indirectly by the states of R.

A *controlled discrete event system* (CDES) is an ordered tuple $Pr_c = (CPN_c, K, \Gamma)$. In a CPN_c, $M[_c(t,b) > M'$ denotes that a marking M' is directly reachable from a marking M by the occurrence of the binding element (t, b) under the control of $\gamma \in \Gamma$. This is equivalent to:

$$\begin{aligned} M[(t,b) > M' \qquad & \text{if } t \in T_u \\ M[(t,b) > M', b \notin \gamma(t) \quad & \text{if } t \in T_c \text{ and } \gamma \in \Gamma \end{aligned}$$

The CDES and controller are then embodied in a closed-loop system to constitute a *supervised discrete event system* (SDES) $S/Pr_c = (Pr_c, S)$. The external control φ is then applied to Pr_c by changing the control pattern through a sequence $\gamma_0, \gamma_1, \ldots$ in a such a way that the $CDES$ behaves in a specified manner. A state of a $CDES$ is a pair (x, M) where $x \in X$ and M is a marking of CPN_c. The behavior of the closed-loop system is defined as follows. If (x, M) is the current state, then the next state is (x', M') if and only if $M[_c(t,b) > M'$, $\xi((t,b), x) = x'$, and $b \notin \varphi(x)(t)$. It is always required that S be *complete* in the sense that $\xi((t,b), x)$ is defined whenever S/Pr_c is in state (x, M) with $M[_c(t,b) > M'$ and $b \notin \varphi(x)(t)$.

We now consider the forbidden state control problem in which the control specification is expressed as a set of forbidden markings $\mathcal{M}_b$. This problem is expressed as follows. Given a set of forbidden markings $\mathcal{M}_b$, an uncontrolled discrete event system (CPN, K), and an initial marking $M_0 \notin \mathcal{M}_b$, the problem is to derive a controller that will prevent the system from reaching any forbidden marking.

Let us consider the forbidden markings for the process described in Fig. 1. The behavior of the trains must be restrained to prevent the trains from colliding. Therefore,

the two trains must be separated by at least one section to ensure that an incoming train can stop at a proper distance. It is assumed that transitions $T4$ and $T6$ are uncontrollable. The set of forbidden markings is formally specified as

$$\begin{aligned}\mathcal{M}_b &= \{1'(S1,Tr_1) + 1'(S1,Tr_2), \ldots, 1'(S6,Tr_1) + 1'(S6,Tr_2) \\ &\quad 1'(S1,Tr_1) + 1'(S2,Tr_2), \ldots, 1'(S6,Tr_1) + 1'(S1,Tr_2)\}\end{aligned}$$

It should be noted that the forbidden markings are not all given explicitly. We give one representative per equivalence class. In our example, equivalence of markings is expressed by a permutation function ψ. A marking M is equivalent to a marking M' if M can be obtained from M' by permuting its colors. For example, a marking in which train number 1 is in section 1 and train number 2 in section 2 is equivalent to a marking in which train number 2 is in section number 1 and train number 1 in section 2.

3 The Synthesis Algorithm

The basic idea behind our control synthesis algorithm is to reduce the number of markings that must be examined by gathering the components that "behave in the same way" into the same equivalence class. The algorithm is based on the notion of *last controllable binding element.* It calculates a compact representation of the controller S. The sets of all markings and steps are partitioned into disjoint nonempty equivalence classes. A controller $\overline{S} = (\overline{R}, \overline{\varphi})$ is computed from representative members, one per class of equivalent markings and class of equivalent steps.

The equivalence relations are specified by the person who analyzes the system. These relations must respect the inherent nature of the system. In general, several symmetry types (rotation, permutation, or identity function) can be involved in the same system (for different color sets). It has been proved [6] that, when the relations are soundly defined, the reduction by means of equivalent markings does not result in loss of information.

Definition

Let x, y, and z denote nodes in the reachability tree, and (t, b) a binding element where t is a controllable transition. The *last controllable binding element* (LCBE) of z is a triple (x, t, b) such that:

1. $M_x[(t,b) > M_y$
2. $M_y[(t_1,b_1)(t_2,b_2)\ldots(t_n,b_n) > M_z$ with $t_i \in T_u$, for $i = 1, \ldots, n$.

The LCBE is interpreted as follows (see Fig. 2). The transition t from the node x to node y is controllable whereas transitions from node y to node z are uncontrollable.

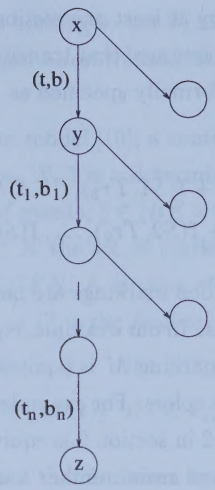


Figure 2: Last controllable binding element of z

Therefore, the only way to make the marking M_z unreachable in the process is by disabling the binding contained in the LCBE of z when the process is in marking M_x. The LCBE of a node is unique since the reachability tree is a tree. In our representation, each node of the reachability tree consists of a marking, a label, and a LCBE.

The algorithm that computes the unique maximal set of admissible markings works as follows. Initially, the node $(M_0, [], ())$ is created and inserted into the set of unprocessed nodes. In the example, $M_0 = 1`(S1, Tr1) + 1`(S4, Tr2)$, the label is empty, and the LCBE is undefined. While there are unprocessed nodes, a node x is selected and processed. The processing of a node starts with a test for an equivalence between the marking M_x and the marking M_y of an already processed node. Only the first-picked node in each equivalence class is developed further. If such a node y exists, the string "equivalent to y" is appended to the label of x (for example, see nodes #8 and #9 in Fig. 3), otherwise every binding element (t, b) enabled in M_x is analyzed. The marking M reached after the occurrence of the binding element (t, b) is computed and checked for an equivalence with some other marking M' included in the set of forbidden markings. If so, there are two cases that is, either t is controllable or not. The first case, in which the transition t is controllable, the binding b is inserted in the set of forbidden bindings of x, that is $\varphi(x)(t)$. For example, the marking $1`(S3, Tr1) + 1`(S4, Tr2)$, reached after the occurrence of the step $(T2, < x = Tr1 >)$ from node #2, is a forbidden marking. Therefore, the binding $< x = Tr1 >$ must be added to $\varphi(\#2)(T2)$. In the second case, in which t is uncontrollable, the feedback function is updated by a recursive procedure briefly illustrated here. Let us consider the subtree of the reachability tree of Fig. 3 as illustrated in Fig. 4. The marking $1`(S2, Tr1) + 1`(S1, Tr2)$, reached after the occurrence of the step

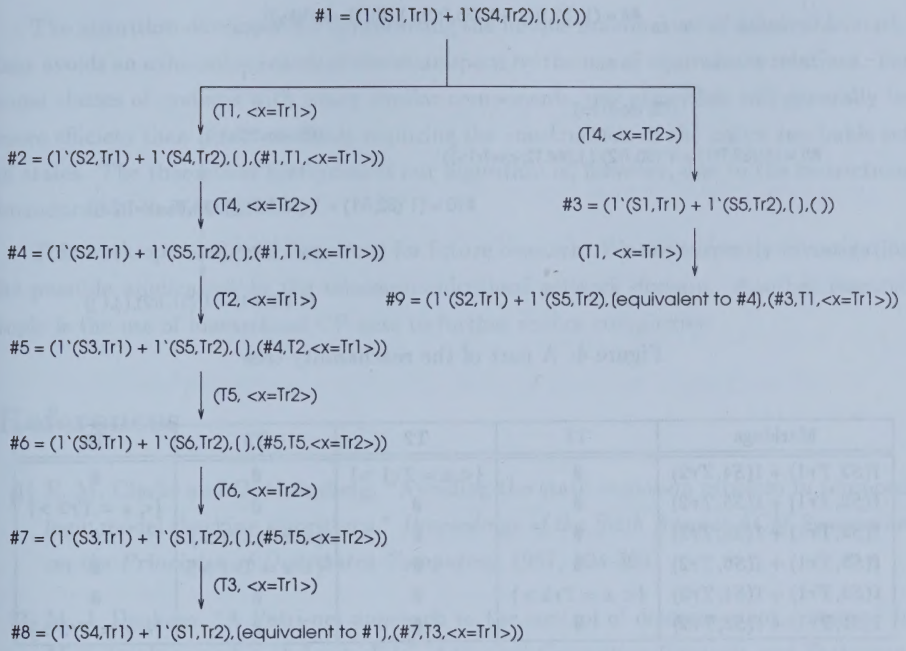


Figure 3: The $\overline{R}$ for the circular railway system

$(T6, < x = Tr2 >)$ from node #10, is a forbidden marking. Since the transition $T6$ is uncontrollable, the controller cannot prevent the step from occurring in the process at this level. Based on the information in $LCBE_{\#10}$, the binding $< x = Tr2 >$ is inserted into $\varphi(\#4)(T5)$.

If M is not equivalent to some forbidden marking in $\mathcal{M}_b$, then the algorithm checks if marking M is equivalent to a marking M_y where y is a son of x. If this condition is satisfied, the string "(t, b)" is appended to the label of the arc from the node x to the node y. That is, information about the fact that a marking equivalent to M_y can be reached from the node x on the occurrence of binding element (t, b) is stored in the label of the arc from node x to node y. The marking M is not considered further.

If all the above conditions are not satisfied, it means that the marking M is not equivalent to a marking of a son of x and not forbidden (for example, nodes #1 to #7 in Fig. 3). In this case, a new node z is created with the following attributes: the marking M, an empty label, and the LCBE_z which is defined as (x, t, b) if the transition t is controllable otherwise the LCBE_x of its parent node x. Furthemore, the set of unprocessed nodes is updated and a new arc is created from node x to node z. Finally, the node x is included in the set of processed nodes and removed from the set of unprocessed nodes.

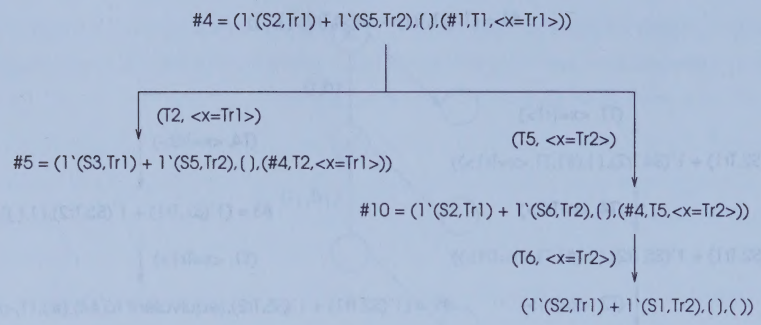


Figure 4: A part of the reachability tree

Markings	T1	T2	T3	T5
$\Gamma(S2,Tr1) + \Gamma(S4,Tr2)$	$\emptyset$	$\{< x = Tr1 >\}$	$\emptyset$	$\emptyset$
$\Gamma(S2,Tr1) + \Gamma(S5,Tr2)$	$\emptyset$	$\emptyset$	$\emptyset$	$\{< x = Tr2 >\}$
$\Gamma(S3,Tr1) + \Gamma(S5,Tr2)$	$\emptyset$	$\emptyset$	$\{< x = Tr1 >\}$	$\emptyset$
$\Gamma(S3,Tr1) + \Gamma(S6,Tr2)$	$\emptyset$	$\emptyset$	$\{< x = Tr1 >\}$	$\emptyset$
$\Gamma(S3,Tr1) + \Gamma(S1,Tr2)$	$\{< x = Tr2 >\}$	$\emptyset$	$\emptyset$	$\emptyset$
$\Gamma(S1,Tr1) + \Gamma(S5,Tr2)$	$\emptyset$	$\emptyset$	$\emptyset$	$\{< x = Tr2 >\}$

Table 1: The feedback function $\overline{\varphi}$

Fig. 3 shows the final solution for the circular-railway example. Less than half of the possible nodes have been inserted in the reachability tree. Furthermore, nodes leading to forbidden markings on uncontrollable transitions have been also removed with their incoming and outgoing transitions. The feedback function $\overline{\varphi}$ is given in Table 1. Since the label and LCBE of each node are useless in the control loop system, only the marking of each node is kept in the solution.

During execution, the compact form $\overline{S}$ and the current marking (state) of S are used to compute the next marking and control pattern. Let M be the current marking, (t,b) the next step, and M' a marking of $\overline{S}$. Let ψ be a function such that $\psi(M) = M'$. Then, $M[(t,b) > \psi^{-1}(M'')$, where $M'[\psi(t,b) > M''$. Furthermore, $\varphi(\psi^{-1}(M''))(t) = \psi^{-1}(\varphi(M'')(t))$.

4 Conclusion

In this paper, we have presented a colored Petri-net approach to the control of discrete event systems. One of the benefits of using CP-nets instead of equivalent PT-nets is the more compact and readable representation of the system. Another benefit is that the use of equivalence relations reduces the cardinality of the set of forbidden states.

The algorithm developed for synthesizing the unique maximal set of admissible markings avoids an exhaustive search of the state space by the use of equivalence relations. For some classes of systems with many similar components, our algorithm will generally be more efficient than direct methods requiring the construction of the entire reachable set of states. The theoretical limitation of our algorithm is, however, due to the restrictions introduced in Section 2.

This work opens several directions for future research. We are currently investigating its possible application to the telecommunications network domain. Another research topic is the use of hierarchical CP-nets to further reduce complexity.

References

[1] E. M. Clarke and O. Grümberg, "Avoiding the state explosion problem in temporal logic model checking algorithms," *Proceedings of the Sixth Annual ACM Symposium on the Principles of Distributed Computing*, 1987, 294-303.

[2] M. J. Denham, "A Petri-net approach to the control of discrete-event systems," in M. J. Denham and A. J. Laub, Eds., *Advanced Computing Concepts and Techniques in Control Engineering*, NATO ASI Series, Vol. F47, Berlin: Springer-Verlag, 1988, 191-214.

[3] A. Giua and F. DiCesare, "Supervisory design using Petri nets," *Proceedings of the 30th Conference on Decision and Control*, 1991, 92-97.

[4] L. E. Holloway and B. H. Krogh, "Synthesis of feedback control logic for a class of controlled Petri nets," *IEEE Transactions on Automatic Control*, 35 (5), 1990, 514-523.

[5] L. E. Holloway and X. Guan, "A generalization of state avoidance policies for controlled Petri nets," Technical Report T93002, Technology from the Center for Robotics and Manufacturing Systems, September 1993.

[6] P. Huber, A. M. Jensen, L. O. Jepsen, and K. Jensen, "Reachability tree for high-level Petri nets," *Theoretical Computer Science*, 45, 1986, 261-292.

[7] K. Jensen, *Coloured Petri Nets, Basic Concepts, Analysis Methods and Practical Use*, Volume 1, Berlin: Springer-Verlag, 1992.

[8] M. D. Jeng and F. DiCesare, "A synthesis method for Petri net modeling of automated manufacturing systems with shared resources," *Proceedings of the 31st Conference on Decision and Control*, 1992, 1184-1189.

[9] B. H. Krogh, "Controlled Petri nets and maximally permissive feedback logic," *Proceedings of the 25th Annual Allerton Conference on Communication, Control, and Computing*, 1987, 317-326.

[10] P. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event processes," *SIAM Journal Control and Optimization*, 25 (1), 1987, 206-230.

[11] R. S. Sreenivas and B. H. Krogh, "On Petri net models of infinite state supervisors," *IEEE Transactions on Automatic Control* 37 (2), 1992, 274-277.

[12] T. Ushio, "Maximally permissive feedback and modular control synthesis in Petri nets with external inputs places," *IEEE Transactions on Automatic Control* 35 (7), 1990, 844-848.

Sensor Assignment for Invertibility of Timed Petri Nets

Yongseok Park and Edwin. K. P. Chong
School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907-1285
e-mail: yongseok@ecn.purdue.edu, echong@ecn.purdue.edu

Abstract

We view a Timed Petri Net (TPN) as an input-output system, where the input is the set of transition lifetimes and the output is the resulting sequence of transitions, states, and transition epochs. The TPN is observed via an observation map, which consists of place sensors and transition sensors. We consider the goal of extracting transition life times from observations of the output of the TPN, referred to as inversion. In this paper, we investigate the problem of designing an observation scheme by assigning a minimal set of sensors to the system to achieve invertibility.

1 Introduction

We view a Timed Petri Net (TPN) as an input-output system, where the input is the set of transition lifetimes and the output is the resulting sequence of transitions, states, and transition epochs. In this context, we consider the goal of extracting transition lifetimes from observations of the output of the TPN. If this goal can be achieved, we say that the system is *invertible*. Invertibility depends on the structure of the TPN and the output observation map.

The above problem has been motivated and studied in several previous papers, for automata models [1], [2], as well as TPN models [3], [4]. Some related work has also been considered in the context of untimed automata [5], fault diagnosis [6], and perturbation tracking [7]. In this paper, we provide several extensions to the results of [3]. Specifically, we generalize the transition observation map and incorporate a *time extraction assumption* into the necessary and sufficient conditions for invertibility. We also consider the problem of assigning transition and place "sensors" to the system such that the property of invertibility is guaranteed. We provide an efficient algorithm to achieve *minimal* sensor configurations for invertibility.

2 System model

This work was supported in part by a grant from the Purdue NSF Engineering Research Center for Intelligent Manufacturing, by the Engineering Foundation through grant RI-A-93-09, and by a Whirlpool Faculty Fellowship.

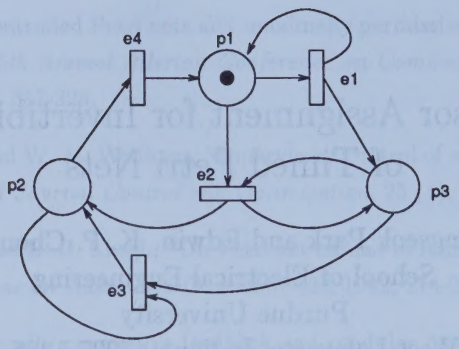


Figure 1: A Petri Net

We consider a Petri Net represented by the 4-tuple $\mathcal{G} = (P, \Sigma, W, m(0))$, where

- $P = \{p_1, p_2, \cdots, p_{|P|}\}$ is a finite set of places;
- $\Sigma = \{\sigma_1, \sigma_2, \cdots, \sigma_{|\Sigma|}\}$ is a finite set of transitions;
- $W : (P \times \Sigma) \cup (\Sigma \times P) \to \mathbb{Z}_+$ is a weight function; and
- $m(0) = [m_1(0), m_2(0), \cdots, m_{|P|}(0)]$ is an initial marking (or state),

where $\mathbb{Z}_+$ denotes the set of nonnegative integers, and $m_i(0)$ is the number of initial tokens in place p_i. It is customary to represent the structure of $\mathcal{G}$ by a Petri Net graph (see, e.g., Figure 1). For background on Petri Nets, see [8].

Let $I(\sigma) = \{p \in P : W(p, \sigma) > 0\}$ denote the set of *input places* and $O(\sigma) = \{p \in P : W(\sigma, p) > 0\}$ the set of *output places* of transition σ. The state transition of the Petri Net $\mathcal{G}$ is characterized by the *next-state function* $\delta : \Sigma \times \mathbb{Z}_+^{|P|} \to \mathbb{Z}_+^{|P|}$. The next-state function δ is a partial function, i.e., for each state $m \in M$, the domain of $\delta(\cdot, m)$ is a subset $\Sigma^F(m) \subset \Sigma$, called the *enabled transition set*. The set $\Sigma^F(m)$ is given by

$$\Sigma^F(m) = \{\sigma \in \Sigma : \forall p_i \in I(\sigma), W(p_i, \sigma) \leq m_i\}.$$

The next state $\delta(\sigma, m)$ by transition σ at state m is given by

$$[\delta(\sigma, m)]_i = m_i - W(p_i, \sigma) + W(\sigma, p_i).$$

2.1 Timed Behavior

A Timed Petri Net (TPN) is defined by its structure $\mathcal{G}$ and a timing mechanism (see dynamics equation below). The behavior of a TPN is determined by a set of positive real numbers, called the *transition lifetimes*. Let $\Delta = \{\Delta(i, \sigma) \in \mathbb{R}^+ : i \in \mathbb{N}, \sigma \in \Sigma\}$ denote the transition lifetime set, where $\mathbb{R}^+$ and $\mathbb{N}$ represent positive real numbers and natural numbers, respectively.

For simplicity, we assume that $\mathcal{G}$ is *persistent*, i.e., for any two enabled transitions, the firing of one transition will not disable the other [9].

The time evolution of the TPN $\mathcal{G}$ is characterized by the following set of equations.

Dynamics equations for Timed Petri Net

Initialization:

- $t(0) = 0$;
- for each $\sigma \in \Sigma$, set $N(\sigma, 0) = 0$;
- for each $\sigma \in \Sigma^F(m(0))$, set $r(\sigma, 0) = \Delta(1, \sigma)$.

Recursion: for $k \geq 0$

$$
\begin{aligned}
d(k+1) &= \min_{\sigma \in \Sigma^F(m(k))} \{r(\sigma, k)\} \\
t(k+1) &= t(k) + d(k+1) \\
\sigma(k+1) &= \arg\min_{\sigma \in \Sigma^F(m(k))} \{r(\sigma, k)\} \\
m(k+1) &= \delta(\sigma(k+1), m(k)) \\
N(\sigma, k+1) &= \begin{cases} N(\sigma, k) + 1 & \text{if } \sigma = \sigma(k+1) \\ N(\sigma, k) & \text{otherwise} \end{cases} \\
\Sigma^{NF}(m(k), \sigma(k+1)) &= \Sigma^F(m(k+1)) - (E^F(m(k)) - \{\sigma(k+1)\}) \\
r(\sigma, k+1) &= \begin{cases} \Delta(N(\sigma, k) + 1, \sigma) & \text{if } \sigma \in \Sigma^{NF}(m(k), \sigma(k+1)) \\ r(\sigma, k) - d(k+1) & \text{if } \sigma \in \Sigma^F(m(k)) - \Sigma^{NF}(m(k), \sigma(k+1)) \end{cases}
\end{aligned}
$$

In the above equations, $\sigma(k)$ denotes the kth transition to fire in the system, $m(k)$ the state just after $\sigma(k)$ fires, and $t(k)$ the time when transition $\sigma(k)$ fires. The quantity $r(\sigma, k)$ represents the *residual lifetime* of transition σ at $t(k)$. The quantity $d(k+1)$ denotes the inter-transition time between $\sigma(k)$ and $\sigma(k+1)$, which is the minimum of the residual lifetimes $r(\sigma, k)$ of all the enabled transitions at state $m(k)$. If the transition σ is enabled at time $t(k)$, it fires at time $t(k) + r(\sigma, k)$. The *score function* $N(\sigma, k)$ denotes the number of firings of transition σ up to $t(k)$. The *new enabled transition set* $\Sigma^{NF}(m(k), \sigma(k+1))$ is the set of transitions that are assigned new lifetimes after transition $\sigma(k+1)$ fires at state $m(k)$.

We view the TPN as an input-output system where the input is the transition lifetime set and the output is the resulting sequence of transitions, states, and transition epochs. Specifically, let $X(n) = \{(\sigma(k), m(k), t(k)) : k \in [1, n]\}$ denote the sequence of transitions, states, and transition epochs up to nth transition, where $[i, j]$ denotes the integer interval from i to j. The output trajectory of the TPN is represented by the sequence $X(n)$.

2.2 Observation Scheme

We need to distinguish between transition observations and place observations. Transitions and places can be either observable or unobservable. If a transition is unobservable, we cannot observe its firing. If it is observable, we observe an output symbol if it fires. Several observable transitions may have the same output symbol. If a place is unobservable, we cannot determine its marking, while if it is observable, we can.

Let $\Lambda_\Sigma : \Sigma \to \Gamma \cup \{\varepsilon\}$ denote the *transition observation map*, where Γ is the set of transition observations and ε represents the *null observation*. Let $\Lambda_{M,i} : \mathbb{Z}_+ \to \mathbb{Z}_+ \cup \{\varepsilon\}$ denote the *place observation map* for place $p_i \in P$. Let P_o denote the set of *observable places*, and P_{uo} the set of *unobservable places*, given by $P_{uo} = P - P_o$. The place observation map is defined by

$$
\Lambda_{M,i}(m_i) = \begin{cases} m_i & \text{if } p_i \in P_o \\ \varepsilon & \text{otherwise.} \end{cases}
$$

Let $\Lambda_M(m) = [\Lambda_{M,1}(m_1), \cdots, \Lambda_{M,|P|}(m_{|P|})]$ denote the *state observation* of state m. Then we represent the *output observation map* Λ at transition epoch $t(k)$ as

$$\Lambda(\sigma(k), m(k), t(k)) = \begin{cases} (\Lambda_\Sigma(\sigma(k)), \Lambda_M(m(k)), t(k)) & \text{if } \Lambda_\Sigma(\sigma(k)) \neq \varepsilon \\ & \quad \text{or } \Lambda_M(m(k)) \neq \Lambda_M(m(k-1)) \\ \varepsilon & \text{otherwise.} \end{cases}$$

Strictly speaking, $\Lambda(\sigma(k), m(k), t(k))$ is a function not just of $\sigma(k)$, $m(k)$, and $t(k)$, but also of $m(k-1)$. The kth output $(\sigma(k)), \Lambda_M(m(k)), t(k))$ is not observed (i.e., is ε) if the observation of transition $\sigma(k)$ is ε, and the state observation does not change when $\sigma(k)$ fires. Otherwise, the output is observed, in the sense that the transition time $t(k)$ can be determined. The above output observation map is different from that of [3]; we feel that the present form is more physically meaningful.

3 Inversion

Given a TPN $\mathcal{G}$ and an observation map Λ as defined previously, we consider the problem of determining the event lifetimes from observations of the output. As in [3], we define the invertibility of a TPN as follows. First, let $Y(n)$ denote the *observation sequence* up to nth transition, given by

$$Y(n) = \{\Lambda(\sigma(k), m(k), t(k)) : k \in [1, n]\}.$$

Invertibility depends on $\mathcal{G}$ and Λ. For this reason, we refer to the pair $(\mathcal{G}, \Lambda)$ as the *system*, and speak of whether or not $(\mathcal{G}, \Lambda)$ is invertible.

Definition 1: The system $(\mathcal{G}, \Lambda)$ is said to be *invertible* if there exists a map A such that for all inputs Δ and $n \in \mathbb{N}$, we have $A(Y(n)) = \{\Delta(i, \sigma) : \sigma \in \Sigma, i \in [1, N(\sigma, n)]\}$.

Our goal in the remainder of this section is to give necessary and sufficient conditions on $\mathcal{G}$ and Λ for invertibility. To this end, let $\sim$ denote the set difference operator, i.e., $C \sim B = (C - B) \cup (B - C)$ for sets B, C. We will need the following definition, taken from [3].

Definition 2: A pair of distinct transitions α, β is said to be an *indistinguishable transition pair* if

1. $\Lambda_\Sigma(\alpha) = \Lambda_\Sigma(\beta)$;
2. $(I(\alpha) \sim I(\beta)) \subseteq P_{uo}$;
3. $(O(\alpha) \sim O(\beta)) \subseteq P_{uo}$;
4. $W(p, \alpha) = W(p, \beta)$, $\forall p \in (I(\alpha) \cap I(\beta) - O(\alpha) \cap O(\beta)) \cap P_o$; and
5. $W(\alpha, p) = W(\beta, p)$, $\forall p \in (O(\alpha) \cap O(\beta) - I(\alpha) \cap I(\beta)) \cap P_o$; and
6. $W(\alpha, p) - W(p, \alpha) = W(\beta, p) - W(p, \beta)$, $\forall p \in I(\alpha) \cap I(\beta) \cap O(\alpha) \cap O(\beta) \cap P_o$.

Let $\mathfrak{I}$ denote the set of all indistinguishable transition pairs. If $(\alpha, \beta) \in \mathfrak{I}$, then $\alpha, \beta \in \Sigma^F(m) \Rightarrow \Lambda_M(\delta(\alpha, m)) = \Lambda_M(\delta(\beta, m))$, i.e., observations of the next states are the same.

In the context of [3], it was shown that the invertibility problem for TPN turns out to be a special *coverability* problem[1]. A similar result holds here. To this end, let $m^{\alpha \vee \beta}$ denote the minimal marking for α and β to be enabled at the same time, given by

$$m_i^{\alpha \vee \beta} = \begin{cases} \max\{W(p_i, \alpha), W(p_i, \beta)\} & p_i \in I(\alpha) \cup I(\beta) \\ 0 & \text{otherwise.} \end{cases}$$

We are now ready to give necessary and sufficient conditions for invertibility.

Theorem 1: The system $(\mathcal{G}, \Lambda)$ is invertible if and only if

1. $\forall \sigma \in \Sigma$, $\Lambda_\Sigma(\sigma) = \varepsilon \Rightarrow \exists p \in P_o$ such that $W(\sigma, p) \neq W(p, \sigma)$; and
2. $\forall (\alpha, \beta) \in \mathfrak{I}$, $m^{\alpha \vee \beta}$ is not coverable in $\mathcal{G}$.

Proof: Condition 1 above is a *time extraction* condition, in that it ensures that $t(k)$ can be determined for each k. Condition 2 ensures that any two indistinguishable transitions cannot be enabled at the same time. In [1], it is proven that the system is invertible if and only if Condition 2 holds with the assumption that every transition epoch is available–time extraction assumption. Therefore, what is left to be proven is the equivalence of Condition 1 and the time extraction assumption.

Suppose Condition 1 holds, then for any transition $\sigma \in \Sigma$ with $\Lambda_\Sigma(\sigma) = \varepsilon$, there exists an observable place p, whose marking changes with the occurrence of σ. Therefore, the transition epoch is available by the observation map.

On the other hand, suppose that every transition epoch is available. Then, by the formulation of the observation map, for any transition $\sigma \in \Sigma$ with $\Lambda_\Sigma(\sigma) = \varepsilon$ and any state $m \in M$, we have $\Lambda_M(m) \neq \Lambda_M(\delta(\sigma, m))$. That implies Condition 1. ∎

4 Sensor Assignment

4.1 Sensors

In the foregoing discussion, we assumed that the observation map is given. In practice, the observation map is determined by physical devices used in a given system to monitor its behavior. We refer to such devices collectively as *sensors*. Our objective here is to develop a model that specifically incorporates the idea of sensors for observation of the TPN.

We consider two kinds of sensors: *place sensors* and *transition sensors*. We can think of a place sensor as a device that can trace the markings of a place. For example, for a place that models a buffer between machines in a manufacturing system, the place sensor may be a counter for parts in the buffer. On the other hand, a transition sensor detects the firing of a transition. For example, the transition sensor may be a signal detector for the request/acknowledgment of a packet in a communication network.

For a given system modeled by a TPN, we assume that there is a pool of sensors that can be deployed in the system. Let $\Psi = \Psi_\Sigma \cup \Psi_P$ denote the pool of available sensors, where Ψ_Σ is a set of transition sensors, and Ψ_P is a set of place sensors.

[1]To define coverability, let $R(m(0))$ denote the set of all reachable states from the initial state $m(0)$. A marking m is said to be *coverable* if there exists $m' \in R(m(0))$ such that $m'_i \geq m_i$, $i \in [1, |P|]$ (see [9]).

We assume that the transition sensors are "binary" in the sense that their outputs are either 0 or 1. Each transition sensor can detect the firing of one or more transitions. If a transition that a sensor can detect fires, then its output is 1; otherwise, its output is 0. Let $f_\Sigma : \Psi_\Sigma \to 2^\Sigma$ denote the *transition sensor map*, which specifies the set of transitions whose firing each sensor can detect.

Each place sensor allows us to keep track of the marking of one or more places. Let $f_P : \Psi_P \to 2^P$ denote the *place sensor map*, which specifies, for each place sensor, the set of places whose marking the sensor can keep track of.

Let $\tilde{\Psi} = \tilde{\Psi}_\Sigma \cup \tilde{\Psi}_P \subset \Psi$ denote the set of sensors deployed in the system. Basically, if $\psi_1 \in \tilde{\Psi}_\Sigma$ and $\sigma \in f_\Sigma(\psi_1)$, then the firing of σ can be detected. Similarly, if $\psi_2 \in \tilde{\Psi}_P$ and $p \in f_P(\psi_2)$, then we can keep track of the marking of p. The set $\tilde{\Psi}$ induces an observation map Λ in a natural way, as we shall discuss in the next section.

4.2 Relation between Sensors and Observation Scheme

Given a set $\tilde{\Psi}$ of sensors deployed in the system, we define an observation map Λ as follows. First, we assume that $\tilde{\Psi}_\Sigma$ is an ordered set, and denote by $\tilde{\psi}_{\Sigma,i}$ its ith element. The transition observation map $\Lambda_\Sigma : \Sigma \to \{0,1\}^{|\tilde{\Psi}_\Sigma|}$ is given by

$$\Lambda_\Sigma(\sigma) = [\Lambda_{\Sigma,1}(\sigma), \cdots, \Lambda_{\Sigma,|\tilde{\Psi}_\Sigma|}(\sigma)],$$

where

$$\Lambda_{\Sigma,i}(\sigma) = \begin{cases} 1 & \text{if } \sigma \in f_\Sigma(\tilde{\psi}_{\Sigma,i}) \\ 0 & \text{otherwise.} \end{cases}$$

We take $\varepsilon = [0, \cdots, 0]$. Similarly, the state observation map is given by

$$\Lambda_M(m) = [\Lambda_{M,1}(m_1), \cdots, \Lambda_{M,|P|}(m_{|P|})],$$

where the place observation map for each $p_i \in P$ is given by

$$\Lambda_{M,i}(m_i) = \begin{cases} m_i & \text{if } \exists \psi \in \tilde{\Psi}_P \text{ s.t. } p_i \in f_P(\psi) \\ \varepsilon & \text{otherwise.} \end{cases}$$

Given Λ_Σ and Λ_M, the output observation map Λ is then defined as before.

4.3 Sensor Assignment for Invertibility

Given a pool of sensors Ψ and a set of deployed sensors $\tilde{\Psi}$, we can determine whether or not the resulting observed system $(\mathcal{G}, \Lambda)$ is invertible. For convenience, we use the notation $(\mathcal{G}, \tilde{\Psi})$ for the observed system in the remainder of the paper.

The following result gives the necessary and sufficient conditions for invertibility in terms of assigned sensors. Let $\oplus$ denote the exclusive-or operation.

Theorem 2: The system $(\mathcal{G}, \tilde{\Psi})$ is invertible if and only if

1. $\forall \sigma \in \Sigma, \exists \psi \in \tilde{\Psi}_\Sigma$ s.t. $\sigma \in f_\Sigma(\psi)$; or $\exists \psi \in \tilde{\Psi}_M, \exists p \in f_M(\psi)$ s.t. $W(\sigma, p) \neq W(p, \sigma)$; and

2. $\forall (\alpha, \beta)$ s.t. $m^{\alpha \vee \beta}$ is coverable in $\mathcal{G}$,

   (a) $\exists \psi \in \tilde{\Psi}_\Sigma$ s.t. $\alpha \in f_\Sigma \oplus \beta \in f_\Sigma$; or

(b) $\exists \psi \in \tilde{\Psi}_P$ s.t. $\exists p \in (I(\alpha) \sim I(\beta)) \cap f_P(\psi)$; or

(c) $\exists \psi \in \tilde{\Psi}_P$ s.t. $\exists p \in (O(\alpha) \sim O(\beta)) \cap f_P(\psi)$; or

(d) $\exists \psi \in \tilde{\Psi}_P$, $\exists p \in (I(\alpha) \cap I(\beta) - O(\alpha) \cap O(\beta)) \cap f_P(\psi)$ s.t. $W(p,\alpha) \neq W(p,\beta)$; or

(e) $\exists \psi \in \tilde{\Psi}_P$, $\exists p \in (O(\alpha) \cap O(\beta) - I(\alpha) \cap I(\beta)) \cap f_P(\psi)$ s.t. $W(\alpha,p) \neq W(\beta,p)$; or

(f) $\exists \psi \in \tilde{\Psi}_P$, $\exists p \in I(\alpha) \cap I(\beta) \cap O(\alpha) \cap O(\beta) \cap f_P(\psi)$ s.t. $W(\alpha,p) - W(p,\alpha) \neq W(\beta,p) - W(p,\beta)$.

Proof: Straightforward by Theorem 1 and the relation between sensors and the observation map. ∎

There may be many possible sensor configurations $\tilde{\Psi}$ that results in an invertible system. In this case, it is desirable to choose $\tilde{\Psi}$ in an "optimal" way. Our criterion for optimality is given in the following definition.

Definition 3: A set of sensors $\tilde{\Psi}$ is said to be a *minimal sensor configuration for invertibility* if $(\mathcal{G}, \tilde{\Psi})$ is invertible but $(\mathcal{G}, \tilde{\Psi} - \psi)$ is not invertible for any $\psi \in \tilde{\Psi}$

Based on Theorem 2, the following is an efficient algorithm for finding minimal sensor configurations for invertibility. Let $\otimes$ denote the pairwise element-set-union operation between families, defined by $A \otimes B = \{a \cup b : a \in A \text{ and } b \in B\}$. For example, $\{\{a\},\{b,c\}\} \otimes \{\{a,b\},\{b,c\}\} = \{\{a,b\},\{b,c\},\{a,b,c\}\}$.

Algorithm 1: *Minimal Sensor Configurations*

$\Xi \leftarrow \{\emptyset\}$

1. For each $\sigma \in \Sigma$
   $\Upsilon \leftarrow \emptyset$
   for each $\psi \in \Psi_\Sigma$
      if $\sigma \in f_\Sigma$ then $\Upsilon \leftarrow \Upsilon \cup \{\psi\}$
   for each $\psi \in \Psi_P$
      for $p \in P$
         if $p \in f_P(\psi)$ and $W(\sigma,p) \neq W(p,\sigma)$ then $\Upsilon \leftarrow \Upsilon \cup \{\psi\}$
   $\Xi \leftarrow \Xi \otimes \Upsilon$
2. For each $(\alpha,\beta) \in \Sigma \times \Sigma$ such that $\alpha \neq \beta$
   if (α,β) is coverable in $\mathcal{G}$ then
      $\Upsilon \leftarrow \emptyset$
      (a) for each $\psi \in \Psi_\Sigma$
         if $\alpha \in f_\Sigma(\psi)$ and $\beta \notin f_\Sigma(\psi)$
            $\Upsilon \leftarrow \Upsilon \cup \{\psi\}$
         if $\alpha \notin f_\Sigma(\psi)$ and $\beta \in f_\Sigma(\psi)$
            $\Upsilon \leftarrow \Upsilon \cup \{\psi\}$
      for each $\psi \in \Psi_P$
         (b) for each $p \in (I(\alpha) \sim I(\beta)) \cap f_P(\psi)$
            $\Upsilon \leftarrow \Upsilon \cup \{\psi\}$
         (c) for each $p \in (O(\alpha) \sim O(\beta)) \cap f_P(\psi)$
            $\Upsilon \leftarrow \Upsilon \cup \{\psi\}$
         (d) for each $p \in (I(\alpha) \cap I(\beta) - O(\alpha) \cap O(\beta)) \cap f_P(\psi)$

if $W(p,\alpha) \neq W(p,\beta)$ then $\Upsilon \leftarrow \Upsilon \cup \{\psi\}$

(e) for each $p \in (O(\alpha) \cap O(\beta) - I(\alpha) \cap I(\beta)) \cap f_P(\psi)$

if $W(\alpha,p) \neq W(\beta,p)$ then $\Upsilon \leftarrow \Upsilon \cup \{\psi\}$

(f) for each $p \in I(\alpha) \cap I(\beta) \cap O(\alpha) \cap O(\beta) \cap f_P(\psi)$

if $W(\alpha,p) - W(p,\alpha) \neq W(\beta,p) - W(p,\beta)$ then $\Upsilon \leftarrow \Upsilon \cup \{\psi\}$

$\Xi \leftarrow \Xi \otimes \Upsilon$

end

The algorithm computes the set of all minimal sensor configurations for invertibility. As it scans each transition and each transition pair, it keeps adding required sensors for invertibility to elements of the set Ξ. After it finishes visiting all transitions and transition pairs, each element of Ξ becomes a minimal sensor configuration for invertibility. If the cost for each sensor is available, we can compare the overall costs of all minimal sensor configurations to choose the optimal configuration.

5 Conclusion

In this paper, we considered the problem of designing observation schemes for the systems modeled by the Time Petri Net, by assigning a minimal set of place sensors and transition sensors to achieve invertibility. We gave a necessary and sufficient condition for invertibility and formulated the sensor schemes for TPNs. Based on the results, we provided an efficient algorithm to compute the set of all minimal sensor configurations. For future research, it is of interest to investigate the sensor assignment problem for "eventual invertibility", which allows a finite delay in inversion.

References

[1] Y. Park and E. K. P. Chong, "Structural sensitivity analysis of discrete event systems by inversion," in *Proc. of 1993 American Control Conference*, (San Francisco, CA, USA), pp. 1214–1218, June 1993.

[2] Y. Park and E. K. P. Chong, "On the eventual invertibility of discrete event systems and its applications," in *Proc. of 32nd Conf. Decision and Control*, (San Antonio, TX, USA), pp. 680–685, 1993.

[3] Y. Park and E. K. P. Chong, "Inversion of timed Petri nets," in *Proc. of 8th IEEE International Symposium on Intelligent Control*, (Chicago, IL, USA), pp. 285–290, 1993.

[4] Y. Park and E. K. P. Chong, "On the eventual invertibility of timed Petri nets," in *Proc. of 36th IEEE Midwest Symposium on Circuits and Systems*, (Detroit, MI, USA), August 1993.

[5] C. M. Özveren and A. S. Willsky, "Invertibility of discrete-event dynamic systems," *Mathematics of Control, Signals, and Systems*, vol. 5, pp. 365–390, 1992.

[6] M. Samprath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Diagnosability of discrete event systems," Tech. Rep. # CGR-94-02, Control Group, College of Engineering, University of Michigan, USA, Mar. 1994.

[7] K. Williams, M. Andersland, J. Gannon, J. Lumpp, and T. Casavant, "Perturbation tracking," in *Proc. of 32nd Conf. Decision and Control*, (San Antonio, Texas, USA), 1993. 674–679.

[8] J. Peterson, *Petri Net Theory and the Modeling of Systems.* Englewood Cliffs: Prentice-Hall Inc., 1981.

[9] T. Murata, "Petri nets: Properties, analysis, and applications," *Proc. of the IEEE*, vol. 77, pp. 541–580, Apr. 1989.

The Use of Coupling in Sensitivity Analysis

Liyi Dai
Department of Systems Science and Mathematics
Washington University, St. Louis, MO 63130

Abstract

In this paper we show that, with the use of the coupling method and in the case of Markov chains, a derivative estimation algorithm converges at rate $N^{-1/2}$ as the simulation length N goes to infinity. We also numerically compare several coupling methods and find that none of them gives dominant performances.

1 Introduction

During the past two decades, the coupling method has developed as an important tool in probability theory. For example, it has been successfully used to prove several important properties of Markov processes such as ergodicity and monotonicity, to find the accuracy of Poisson approximations [11], and to find derivative estimates [1].

More and more evidences show that the coupling method is very useful in sensitivity analysis of discrete event dynamic systems. In this paper, we take Markov chains as the canonical systems to explain the role played by the coupling method in derivative estimation. We prove that appropriate coupling can increase the rate of convergence and reduce the computational effort of several perturbation analysis methods. We also compare different coupling methods for a birth and death process.

1.1 Markov chains

Let $\mathcal{S} = \{x_0, x_1, x_2, ..., x_{d-1}\}$ be a finite state space. Consider a homogeneous discrete-time Markov chain $\mathcal{X} = \{X_n \in \mathcal{S}, n \geq 0\}$ governed by a stochastic matrix $P(\theta) \in R^{d \times d}$, where $\theta \in \Theta = (a, b)$ is a parameter. Each element $p_\theta(x, y) \geq 0, x, y \in \mathcal{S}$, of $P(\theta)$ is the probability of making a state transition from x to y. If $\mathcal{X}$ is irreducible and aperiodic, then there exists a row vector of stationary (steady-state) state distribution $\pi_\theta = [\pi_\theta(x)]$ satisfying

$$\pi_\theta = \pi_\theta P(\theta), \quad \pi_\theta e = 1, \tag{1}$$

where e is a column vector all of whose entries are unity.

For each $x \in \mathcal{S}$, we define the distribution $F_\theta(x, y)$ of making state transitions from x as $F_\theta(x, y) = \sum_{j=0}^{i} p_\theta(x, x_j)$, if $y = x_i, 0 \leq i \leq d-1$. Then $0 \leq F_\theta(x, y) \leq 1$ and $F_\theta(x, x_{d-1}) = 1$. The inverse of $F_\theta(x, y)$ is $F_\theta^{-1}(x, \xi) = \min\{y : F_\theta(x, y) \geq \xi\}$ for any $0 \leq \xi \leq 1$. For each pair $x, y = x_i \in \mathcal{S}$, we define two adjacent states

$$R_-(x, y) = \{x_j : j = \max\{k : k < i, p_\theta(x, x_k) > 0\}$$

$$R_+(x, y) = \{x_j : j = \min\{k : k > i, p_\theta(x, x_k) > 0\}.$$

It is possible that $R_-(x, y)$ and $R_+(x, y)$ are empty for some $x, y \in \mathcal{S}$.

1.2 Performance measures

Let $f(x) : \mathcal{S} \to R$ be a cost (or reward) function of the state being in x. We consider a stationary performance measure $J(\theta) = E_{\pi_\theta}[f(x)] : \Theta \to R$ which is the mean cost in stationary. We assume that $J(\theta)$ is a function of θ through $P(\theta)$ only. The state space $\mathcal{S}$ and the cost function $f(x)$ do not explicitly depend on θ. We are interested in obtaining an estimate of $J'(\theta)$, the derivative of $J(\theta)$. Such derivative information is very useful to, for example, optimization. Since $J(\theta) = \sum_{x\in\mathcal{S}} \pi_\theta(x)f(x)$, we have $J'(\theta) = \sum_{x\in\mathcal{S}} \pi'_\theta(x)f(x)$. For many systems it is infeasible or inefficient to solve (1). We are interested in estimating $J'(\theta)$ from sample path observation or simulation.

If $\mathcal{X}$ is irreducible and aperiodic, then there exists a $\gamma(\theta) > 0$ such that, for any fixed positive integer N, (e.g., [10, p.138])

$$|\frac{1}{N}\sum_{n=1}^{N} E[f(X_n)] - J(\theta)| \leq \frac{\gamma(\theta)}{N}. \tag{2}$$

2 The Coupling Method

Let α_1 and α_2 be two probability measures defined on a same measurable space $(\Omega, \mathcal{F})$. By coupling we mean a probability measure $\tilde{\alpha}$ on $(\Omega \times \Omega, \mathcal{F} \times \mathcal{F})$ such that

$$\alpha_1 = \tilde{\alpha}\beta_1^{-1}, \quad \alpha_2 = \tilde{\alpha}\beta_2^{-1},$$

where $\beta_1(x,y) = x$, $\beta_2(x,y) = y$ for $(x,y) \in \Omega \times \Omega$.

Therefore, roughly speaking, coupling means any correlationship of two random variables (or vectors) with given marginals. The coupling method has been used in many aspects since its invention by Doeblin [4]. Readers are referred to [11] for a complete theory of this method and its use in simulation.

Let X and Y be two (continuous or discrete) random variables on a same state space $S \subset R$ with distributions $\alpha_1(x)$ and $\alpha_2(y)$, respectively. Then there are many ways to generate samples of X and Y, which actually corresponds to different coupling schemes.

2.1 *Maximal correlation*

Let ξ be uniform on $[0,1]$. Then the following generation of X and Y defines a coupling:

$$X = \alpha_1^{-1}(\xi), \quad Y = \alpha_2^{-1}(\xi). \tag{3}$$

Although it satisfies the definition of coupling, (3) is called maximally correlated sampling since it maximizes $Cov(X,Y)$ [9]. If $\alpha_\theta^{-1}(\xi)$ is differentiable in θ with bounded derivative, the sample $X = \alpha_\theta^{-1}(\xi)$ is Lipschitz continuous in θ. The method of Infinitesimal perturbation analysis takes advantage of this property to yield derivative estimates for a large class of systems.

2.2 γ *coupling*

For any coupling between X and Y, we have [11, p.10-12; 1]

$$Prob\{X = Y\} \leq 1 - s, \quad s = \frac{1}{2}\int |\alpha_1(x) - \alpha_2(x)|.$$

Any coupling achieving the upper bound of $Prob\{X = Y\}$ is called γ coupling. The following scheme is a γ coupling: Define four random variables $U \in \{0,1\}$, $V, W, Z \in \mathcal{S}$ such that $Prob\{U = 1\} = 1 - Prob\{U = 0\} = 1 - s$, $Prob\{V = x\} = \min\{\alpha_1(x), \alpha_2(x)\}/(1 - s)$, $Prob\{W = x\} = (\alpha_1(x) - \alpha_2(x))^+/s$, and $Prob\{Z = x\} = (\alpha_2(x) - \alpha_1(x))^+/s$. Let $X = UV + (1 - U)W$ and $Y = UV + (1 - U)Z$. Then, X and Y are γ coupled [11]. In some literature this scheme is also called maximal coupling. However, we reserve the term maximal coupling for a standard coupling scheme of Markov chains that will be introduced in the next subsection.

2.3 Coupling of Markov chains

Let $\mathcal{X} = \{X_n\}$ and $\mathcal{X}' = \{X'_n\}$ be two Markov chains on $\mathcal{S}$ governed by the same $P(\theta)$ but $X_0 \neq X'_0$. We say that a coupling of $\mathcal{X}$ and $\mathcal{X}'$ is successful if there exists an integer $T < \infty$ such that $X_n = X'_n, a.s., \forall\ n \geq T$, and we call T the coupling time. The following are several coupling schemes of Markov chains that we will need.

γ coupling

This scheme refers to the case where, for each $n > 0$, X_n and X'_n are generated according to the γ coupling method. If X_n and X'_n are coupled by the γ coupling of Subsection 2.2, then $T < \infty, a.s.$ To see this point, let us consider a new Markov chain $\tilde{\mathcal{X}} = \{\tilde{X}_n = (X_n, X'_n) \in \mathcal{S} \times \mathcal{S}, n \geq 0\}$. Denote $\mathcal{D} = \{(x,x) : x \in \mathcal{S}\}$. Then $\mathcal{D}$ is obviously a recurrent closed set. Consider any $\tilde{X}_{n-1} = (x,x') \in (\mathcal{S} \times \mathcal{S}) \setminus \mathcal{D}$ and the next state $\tilde{X}_n$ is generated by the γ coupling method of Subsection 2.2. If $s < 1$, then $P\{\tilde{X}_n \in \mathcal{D}\} = P\{U = 1\} = 1 - s > 0$. With a positive probability (x,x') gets absorbed in $\mathcal{D}$, that is, (x,x') is transient. If $s = 1$, the γ coupling method is equivalent to independent simulations of X_n and X'_n starting from x and x'. Since $\{X_n\}$ is irreducible and aperiodic, with a positive probability that $\{\tilde{X}_{n+i}, i \geq 0\}$ will reach a state (y,y') for which $s < 1$. The state (x,x') is also transient in this case. Thus all states in $(\mathcal{S} \times \mathcal{S}) \setminus \mathcal{D}$ are transient. Since a finite Markov chain can only visit a transient state for a finite number of times, $T < \infty, a.s.$

Doeblin coupling

This is the first coupling scheme invented by Doeblin [4]. For every $n > 0$, we independently generate X_n and X'_n until we reach a T, $T < \infty, a.s.$, such that $X_T = X'_T$. Then we set $X_n = X'_n$ for all $n > T$.

Maximal coupling

A coupling scheme is called maximal coupling if it minimizes the coupling time T. Goldstein gave a maximal coupling scheme [8]. However, his scheme uses the information of state distribution at every step n, which is not useful to us. Maximal coupling is an important issue for derivative estimation that warrants further research.

3 Coupling in Perturbation Analysis

Since the whole theory of perturbation analysis is based on the concept of correlating sample pathes at θ and $\theta + \delta$, it satisfies the coupling definition. However, here we are going to investigate a situation where the coupling method deserves special credit. Particularly, [1] has successfully applied the γ coupling method to obtain perturbation analysis estimates for general discrete event dynamic systems.

Let us define a finite-horizon performance measure $J(N,\theta)$ by

$$J(N,\theta) = \frac{1}{N}\sum_{n=1}^{N} E[f(X_n)].$$

We consider constructing an estimator h_N for $J'(N,\theta)$. We start with a special coupling scheme. Let $X_0 \in \mathcal{S}$. At the nth, $n \geq 1$, transition with state X_{n-1}, we generate a random number ξ_n uniform on $[0,1]$ and let the next state be $X_n(\xi_n) = F_\theta^{-1}(X_{n-1},\xi_n)$. Direct verification shows that $\{X_n(\xi_n)\}$ generated in this way has the transition matrix $P(\theta)$. For any small $\delta > 0$, we generate another Markov chain $\{Y_n\}$ governed by $P(\theta+\delta)$ in the following way: Let $Y_0 = X_0$. At the nth transition with state $Y_{n-1} \in \mathcal{S}$, we first generate a random number u_n uniform on $[0,1]$ and define $y = F_\theta^{-1}(Y_{n-1},u_n) \in \mathcal{S}$. Then we use another random number v_n uniform on $[0,1]$ to generate $Y_n(u_n,v_n)$ such that

$$Y_n(u_n,v_n) = \begin{cases} y & \text{with prob. } 1 - \frac{\{-\Delta F_\theta(Y_{n-1},y)\}^+}{p_\theta(Y_{n-1},y)} - \frac{\{\Delta F_\theta(Y_{n-1},R_-(Y_{n-1},y))\}^+}{p_\theta(Y_{n-1},y)} \\ R_+(Y_{n-1},y) & \text{with prob. } \frac{\{-\Delta F_\theta(Y_{n-1},y)\}^+}{p_\theta(Y_{n-1},y)} \\ R_-(Y_{n-1},y) & \text{with prob. } \frac{\{\Delta F_\theta(Y_{n-1},R_-(Y_{n-1},y))\}^+}{p_\theta(Y_{n-1},y)} \end{cases}$$

where $\Delta F_\theta(Y_{n-1},y) = F_{\theta+\delta}(Y_{n-1},y) - F_\theta(Y_{n-1},y)$. Direct verification yields that the Markov chain $\{Y_n(u_n,v_n)\}$ takes $P(\theta+\delta)$ as its transition matrix. Let us couple the generation of $\{X_n(\xi_n)\}$ and $\{Y_n(u_n,v_n)\}$ by setting $u_n = \xi_n$ for $n = 1,2,...$ Then

$$J(N,\theta) = \frac{1}{N}\sum_{n=1}^{N} E_{\{\xi_n\}}[f(X_n(\xi_n))] \text{ and } J(N,\theta+\delta) = \frac{1}{N}\sum_{n=1}^{N} E_{\{\xi_n,v_n\}}[f(Y_n(\xi_n,v_n))].$$

Therefore,

$$\frac{1}{\delta}[J(N,\theta+\delta) - J(N,\theta)] = \frac{1}{N\delta}\sum_{n=1}^{N} E_{\{\xi_n\}}[E_{\{v_n\}}[f(Y_n(\xi_n,v_n))|\{\xi_n\}] - f(X_n(\xi_n))] \quad (4)$$

Taking limits on both sides of (10), we can change the order of the limit and the expectation with respect to $\{\xi_n\}$ on the right hand side to obtain [3]

$$J'(N,\theta) = \frac{1}{N}E_{\{\xi_n\}}\{\sum_{n=1}^{N}[q_n^+\Delta_n^+ + q_n^-\Delta_n^-]\}$$

where

$$q_n^+ = \frac{\{-F'_\theta(X_{n-1},R_+(X_{n-1},X_n))\}^+}{p_\theta(X_{n-1},X_n)}, \quad q_n^- = \frac{\{F'_\theta(X_{n-1},R_-(X_{n-1},X_n))\}^+}{p_\theta(X_{n-1},X_n)}$$

$$\Delta_n^+ = \sum_{i=n}^{N}[f(X_i'(n)) - f(X_i)], \quad \Delta_n^- = \sum_{i=n}^{N}[f(X_i''(n)) - f(X_i)]$$

and $\mathcal{X}'(n) = \{X_i'(n), N \geq i \geq n\}, \mathcal{X}''(n) = \{X_i''(n), N \geq i \geq n\}$ are sample path realizations of Markov chains governed by the same $P(\theta)$ with respective initial states $X_n'(n) = R_+(X_{n-1},X_n)$ and $X_n''(n) = R_-(X_{n-1},X_n)$. Therefore,

$$h_N = \frac{1}{N}\sum_{n=1}^{N}[q_n^+\Delta_n^+ + q_n^-\Delta_n^-] \quad (5)$$

is an unbiased estimate of $J'(N,\theta)$. The condition for the unbiasedness of (5) are very mild. There are other forms of estimates of the type of (5) available [3]. For simplicity, we will proceed with our discussion based on the estimate (5).

Although the conditions for its unbiasedness are very mild, (5) has potential of becoming quite complicated in implementation: For every $n, 0 < n < N$, we need to perform extra simulations of $\mathcal{X}'(n)$ and $\mathcal{X}''(n)$. By the end of the Nth state transition, we may need to simultaneously simulate $2N$ additional chains $\{\mathcal{X}'(n), \mathcal{X}''(n)\}$, which is very time-consuming and undesirable, especially for large N.

Fortunately, such computational effort in the implementation of (5) can be greatly reduced by using the coupling method. For every $0 < n < N$, if $\mathcal{X}'(n)$ and $\mathcal{X}''(n)$ are generated such that both of them are γ coupled with $\mathcal{X}$ in a way presented in Section 2.3, we know that the coupling between $\mathcal{X}'(n)$ (or $\mathcal{X}''(n)$) and $\mathcal{X}$ is guaranteed successful with an a.s. finite coupling time T_n^+ (or T_n^- respectively). Therefore, if γ coupling is used, both $\mathcal{X}'(n)$ and $\mathcal{X}''(n)$ last for only an a.s. finite number of steps. We only need a finite number of additional simulations at each step.

4 Rates of convergence

Assume that the coupling method is used in the implementation of (5) with successful coupling times. Let us rewrite (5) as

$$h_N = \frac{1}{N}\sum_{n=1}^{N} \phi_n, \quad \phi_n = q_n^+ \Delta_n^+ + q_n^- \Delta_n^-, \tag{6}$$

where

$$\Delta_n^+ = \sum_{i=n}^{\min\{n+T_n^+, N\}} [f(X_i'(n)) - f(X_i)], \quad \Delta_n^- = \sum_{i=n}^{\min\{n+T_n^-, N\}} [f(X_i''(n)) - f(X_i)].$$

Next, we investigate the rate of convergence for the derivative estimate (6). We need the following lemma that was due to [2].

Lemma 1. Let s, B, A_n, b_n be real numbers, $s \geq 0$, $B > 0$. Assume that $c = \lim_{n\to\infty} A_n - s$ exists and is finite. If for $n \geq n_0$,

$$b_{n+1} \leq b_n(1 - \frac{A_n}{n}) + \frac{B}{n^{1+s}}$$

and if $c > 0$, then

$$\lim_{n\to\infty} \sup n^s b_n \leq B/c.$$

Lemma 2. If the chain $\mathcal{X}$ is irreducible and aperiodic, then

$$J'(N,\theta) - J'(\theta) = O(N^{-1}).$$

Proof. The performance measure $J(N,\theta)$ can be written as

$$J(N,\theta) = \frac{1}{N}\sum_{n=1}^{N} E[f(X_n)] = \frac{1}{N}\sum_{n=1}^{N} \pi^{(0)} P^n(\theta) F,$$

where $\pi^{(0)}$ is the state distribution at $n = 0$ and $F = [f(x_0), f(x_1), ..., f(x_{d-1})]^T$ is the column vector of cost function. Then

$$J(N,\theta) - J(\theta) = \frac{1}{N}\sum_{n=1}^{N}(\pi^{(0)}P^n(\theta) - \pi_\theta)F = \frac{1}{N}\sum_{n=1}^{N}\pi^{(0)}(P^n(\theta) - e\pi_\theta)F.$$

Since $P(\theta)e = e$, direct verification yields $P^n(\theta) - e\pi_\theta = (P(\theta) - e\pi_\theta)^n$. Therefore,

$$\begin{aligned} J(N,\theta) - J(\theta) &= \frac{1}{N}\sum_{n=1}^{N}\pi^{(0)}(P(\theta) - e\pi_\theta)^n F \\ &= \frac{1}{N}\pi^{(0)}(P(\theta) - e\pi_\theta)Z(\theta)[I - (P(\theta) - e\pi_\theta)^N]F, \end{aligned}$$

where $Z(\theta) = (I - P(\theta) + e\pi_\theta)^{-1}$ is the fundmental matrix. Thus

$$\begin{aligned} J'(N,\theta) - J'(\theta) &= \frac{1}{N}\pi^{(0)}((P(\theta) - e\pi_\theta)Z(\theta))'[I - (P(\theta) - e\pi_\theta)^N]F \\ &\quad - \pi^{(0)}(P(\theta) - e\pi_\theta)Z(\theta)(P(\theta) - e\pi_\theta)^{N-1}(P'(\theta) - e\pi'_\theta)F. \end{aligned}$$

For irreducible finite-state Markov chains, $(P(\theta) - e\pi)^N$ goes to zero exponentially as N goes to infinity [5, p.173]. We thus conclude from the previous equation that $J'(N,\theta) - J'(\theta) = O(N^{-1})$. □

Lemma 3. Assume that $\mathcal{X}$ is irreducible and aperiodic. If the γ coupling method is used in (6), then the random variables T_n^+ and T_n^- do not depend on n. Moreover, there exists a random variable T such that $E[T] < \infty$ and

$$T_n^+ \le T, \quad T_n^- \le T, \quad a.s.$$

Proof. It is obvious from the Markovian property that T_n^+ and T_n^- do not depend on n. They only depend on X_n, $R_+(X_{n-1}, X_n)$, and $R_-(X_{n-1}, X_n)$. Also we know from the theory of coupling that $E[T_n^+] < \infty$ and $E[T_n^+] < \infty$ since $\mathcal{X}$ is a finite-state Markov chain [10, p.103]. If we choose

$$T = \sum_{\{X_n, R_+(X_{n-1},X_n), R_-(X_{n-1},X_n)\}} (T_n^+ + T_n^-),$$

the lemma follows naturally. □

Theorem 1. Assume that the chain $\mathcal{X}$ is irreducible and aperiodic, the γ coupling method used in (6), $|q_n^+|$ and $|q_n^-|$ are uniformly bounded by a constant $C > 0$ for any n. Then

$$(E[(h_N - J'(\theta))^2])^{1/2} = O(N^{-1/2}).$$

In other words, h_N is a strongly consistent estimate. Its rate of convergence is $N^{-1/2}$ which is the best possible rate for general estimation algorithms.

Proof. First,

$$\begin{aligned} &E[(h_N - J'(\theta))^2] = Var[h_N] + (E[h_N] - J'(\theta))^2 \\ &= Var[h_N] + (J'(N,\theta) - J'(\theta))^2 = Var[h_N] + O(N^{-2}), \end{aligned} \tag{7}$$

where the last equality comes from Lemma 2.

According to (6), we know that h_N has the following recursive form:

$$h_N = (1 - \frac{1}{N})h_{N-1} + \frac{1}{N}\phi_N.$$

Therefore,

$$Var[h_N] = (1 - \frac{1}{N})^2 Var[h_{N-1}] + \frac{1}{N^2}Var[\phi_N] + \frac{2(N-1)}{N^2}Cov[h_{N-1}, \phi_N]$$

$$= (1 - \frac{1}{N})^2 Var[h_{N-1}] + \frac{1}{N^2}Var[\phi_N] + \frac{2}{N^2}\sum_{n=1}^{N-1} Cov[\phi_n, \phi_N]. \tag{8}$$

According to (6), for all $1 \le n \le N$, $|Cov[\phi_n, \phi_N]|$ is bounded by

$$|Cov[q_n^+\Delta_n^+, q_N^+\Delta_N^+]| + |Cov[q_n^+\Delta_n^+, q_N^-\Delta_N^-]| + |Cov[q_n^-\Delta_n^-, q_N^+\Delta_N^+]| + |Cov[q_n^-\Delta_n^-, q_N^-\Delta_N^-]|. \tag{9}$$

Next we consider the first term in (9). It equals

$$|E[\sum_{i=n}^{\min\{n+T_n^+, N\}} Cov[q_n^+(f(X_i'(n)) - f(X_i)), q_N^+(f(R_+(X_{N-1}, X_N)) - f(X_N))]]|.$$

Under the assumption that $|q_n^+| \le C$ and $|q_n^-| \le C$, there exist a $\xi > 0$ and $0 \le \rho < 1$ such that (Doob 1953, Lemma 7.1, p.222),

$$Cov[q_n^+(f(X_i'(n)) - f(X_i)), q_N^+(f(R_+(X_{N-1}, X_N)) - f(X_N))]] \le \xi\rho^{N-i}$$

for any $n \le i \le N$. Let T be chosen as in Lemma 3. Then we know from the previous inequality that, conditional on all T_n^+, T_n^-, the first term in (9) is bounded by

$$\sum_{i=n}^{\min\{n+T_n^+, N\}} \xi\rho^{N-i} \le \frac{\xi}{1-\rho}\rho^{N-(n+T_n^+)} \le \frac{\xi}{1-\rho}\rho^{N-(n+T)}, \ a.s. \tag{10}$$

for $n < N - T_n^+$. A similar analysis shows that each of the other three terms in (9) is also bounded by the right hand side of (10). Therefore,

$$|\sum_{n=1}^{N-1} |Cov[\phi_n, \phi_N]|| = |Cov[\sum_{n=1}^{N-T} \phi_n + \sum_{n=N-T}^{N} \phi_n, \phi_N]|$$

$$\le \frac{4\xi}{(1-\rho)^2} + |Cov[\sum_{n=N-T}^{N} \phi_n, \phi_N]| \le B_1 < \infty,$$

where B_1 does not depend on N. Since $Var[\phi_N]$ is also bounded, there exists a $B > 0$ such that

$$Var[h_N] \le (1 - \frac{1}{N}(2 - \frac{1}{N}))Var[h_{N-1}] + \frac{B}{N^2}.$$

Setting $A_N = 2 - 1/N$, s=1, c=1, we know immediately from Lemma 1 that $Var[h_N] = O(N^{-1})$. Its substitution into (7) yields the conclusion of the theorem. □

Remark: We use the coupling method in the calculation of Δ_n^+, Δ_n^-. An alternative method was proposed in [6-7] where $\{X_i'(n)\}$ is generated in the following way: $\{X_i'(n)\}$ is independently generated until it reaches the state X_n at $i = T'(X_n)$. Then we set $X'_{i-n+T'(X_n)} = X_i$ for all $i \ge n$. In this case,

$$\Delta_n^+ = \sum_{i=n}^{\min\{N, T'(X_n)\}} f(X_i'(n)) - \sum_{i=\min\{N, T'(X_n)\}}^{N} f(X_i).$$

A comparison on the efficiency of this method and the coupling method is subject to further research.

5 Comparison of coupling methods

There are many coupling schemes. Then which one is the best for the implementation of (5)? The maximal coupling is a natural choice since it minimizes the coupling time. Unfortunately, it is subject to future research to find a maximal coupling scheme that can be used for our problem.

Next we compare different coupling schemes for the implementation of (5). To make the comparison more interesting, as we shall see later, we consider the simulation of a continuous-time birth and death process $\{X_t, t \geq 0\}$ with birth rate θ, $0 < \theta \leq 1$, death rate $\mu = 1 - \theta$, and the maximal population size M. Such a birth and death process can be simulated by the following uniformization procedure: We generate a Poisson point process $\{t_n, n \geq 0\}$ with unity rate and a Markov chain $\{X_n\}$ governed by

$$P(\theta) = \begin{bmatrix} 1-\theta & \theta & & & & \\ 1-\theta & 0 & \theta & & & \\ & \cdots & & & & \\ & & & 1-\theta & 0 & \theta \\ & & & & 1-\theta & \theta \end{bmatrix}.$$

Then $X_t = X_n$ for $t \in [t_n, t_{n+1})$ is a simulation realization of $\{X_t\}$. Consider the performance measure $J(\tau, \theta) = (1/\tau) \int_0^\tau E[f(X_t)]dt$ with τ fixed. Then

$$J(\tau, \theta) = \frac{1}{\tau} E[\sum_{n=0}^{N} f(X_n)(t_{n+1} - t_n)]$$

where N is the total number of Poisson points on $[0, \tau]$ and $t_{N+1} = \tau$. For this example, direct verification shows that (5), which applies with slight modifications, becomes

$$h_N = \frac{1}{\tau\theta} \sum_{n=1}^{N} \Delta_n, \quad \Delta_n = \sum_{i=n}^{N} [f(X_i''(n)) - f(X_i)](t_{i+1} - t_i) \tag{11}$$

where $\{X_i''(n), i \geq n\}$ is a Markov chain governed by $P(\theta)$ starting with $X_n''(n) = X_n - 1$.

Of course, we can couple the simulation of $\{X_n\}$ and $\{X_i''(n)\}$ as the coupling between two discrete-time Markov chains. However, we have an alternative with some very interesting properties, which is the main motivation for us to consider a continuous-time process instead of a discrete-time one. According to (11),

$$\Delta_n = \int_{t_n}^{\tau} [f(X_t''(n)) - f(X_t)]dt$$

where $X_t''(n) = X_i''(n)$, $t \in [t_i, t_{i+1})$, is a same birth and death process with initial state $X_{t_n}''(n) = X_n - 1$. Let us couple $\{X_t''(n)\}$ and $\{X_t\}$ in the following way: Generate a Poisson point process $\{t_k', k \geq 0\}$ with rate 2 and a two-dimensional Markov chain $\{(Y_{1,k}, Y_{2,k}), k \geq 0\}$ with $(Y_{1,0}, Y_{2,0}) = (X_n, X_{n-1})$. When $(Y_{1,k}, Y_{2,k})$ is in (y_1, y_2), the next state is $(\min\{y_1 + 1, M\}, y_2)$, $(\{y_1 - 1\}^+, y_2)$, $(y_1, \min\{y_2 + 1, M\})$, $(y_1, \{y_2 - 1\}^+)$ with respective probabilities $0.5\theta, 0.5\mu, 0.5\theta, 0.5\mu$. Once $Y_{1,k} = Y_{2,k}$, we keep $Y_{1,k} = Y_{2,k}$ afterwards. Then we let $(X_t, X_t''(n)) = (Y_{1,k}, Y_{2,k})$ for $t \in [t_k', t_{k+1}')$. In this case,

$$\Delta_n = \sum_{k=0}^{\tau} [f(Y_{2,k}) - f(Y_{1,k})](t_{k+1}' - t_k').$$

This is actually a Doeblin coupling for continuous-time Markov processes. We call it Doeblin coupling via uniformization, U-D coupling for short, for the ease of discussion.

Therefore, we consider the following four coupling schemes: the Doeblin coupling, the maximal correlation coupling, the γ coupling, and the U-D coupling.

Next, we study the coupling time of $\{X_i''(n)\}$ and $\{X_n\}$, both governed by $P(\theta)$, with $X_0 = x, X_0'' = x - 1$. Let $T(x, \theta)$ denote the mean coupling time. We next examine $T(x, \theta)$ as a function of x and θ for the previously mentioned four coupling schemes.

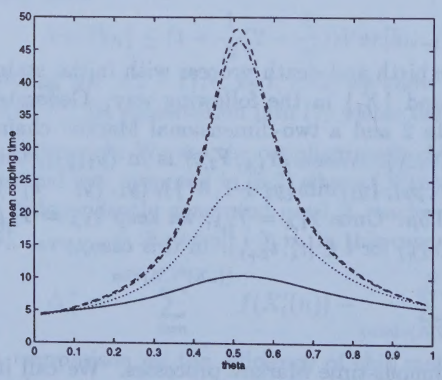
Let $\tilde{P}(\theta)$ be the state transition matrix governing $\tilde{\mathcal{X}} = \{(X_n, X_n'')\}$ under any of the four coupling methods. Since $\mathcal{D} = \{(x, x), x \in \mathcal{S}\}$ is recurrent closed, for each coupling scheme, $\tilde{P}(\theta)$ is of the form

$$\tilde{P}(\theta) = \begin{bmatrix} P(\theta) & 0 \\ Q_1(\theta) & Q_2(\theta) \end{bmatrix}.$$

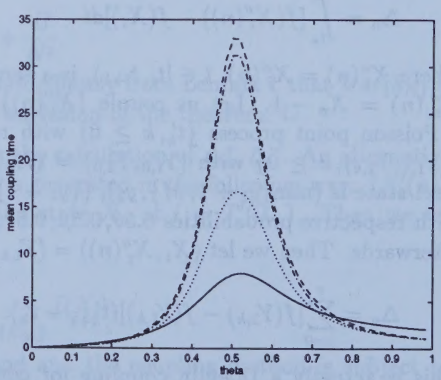
Then the vector of mean time absorbed in $\mathcal{D}$ (the mean coupling time) from states in $(\mathcal{S} \times \mathcal{S}) \setminus \mathcal{D}$ is given by $(I - Q_2(\theta))^{-1} e$ [10, p.103] which can be evaluated numerically for each coupling scheme. We take $M = 9$. The numerical results are displayed in Figure 1, where the "– –" line, the "-.-" line, the "..." line, and the solid line correspond respectively to the Doeblin coupling, the maximal correlation coupling, the γ coupling, and the U-D coupling. In Figure 1 (a) $E_{\pi_\theta}[T(x, \theta)]$ is the mean coupling time over all states. It represents the mean number of coupling steps needed in the implementation of (11).

Figure 1 shows that (i) the γ coupling only slightly better than the Doeblin coupling; (ii) although we may consider the U-D coupling is overall much better than other methods, it can be the worst when θ is small or close to 1; (iii) we cannot conclude that the Doeblin method is the worst since the U-D coupling is its special case; (iv) it seems that the maximal correlation coupling is better than both the Doeblin coupling and the γ coupling. However, if we change (3) by $X = \alpha_1^{-1}(\xi), Y = \alpha_2^{-1}(1 - \xi)$, which corresponds to the incorrect state alignment of X and Y, this method becomes the worst. This situation is illustrated by the dots in figure (d).

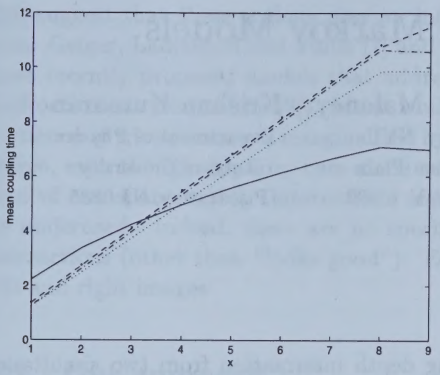
In conclusion, no one method can give dominant performances. Any one coupling method can be the best, depending on specific structures of systems and simulation procedures that we use. It is quite disappointing that we cannot draw a conclusion on choosing an appropriate coupling method for (5). On the other hand, this also provides an opportunity for further research. A challenging problem is to find a coupling scheme better than all the four methods we have discussed here.



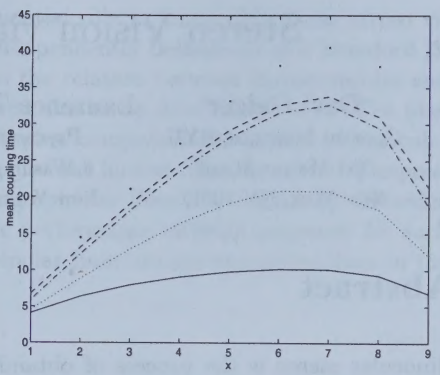
(a) $E_{\pi_\theta}[T(x, \theta)]$



(b) $T(x, \theta)$ for $x = 4$



(c) $T(x,\theta)$ for $\theta = 0.1$



(d) $T(x,\theta)$ for $\theta = 0.4$

Figure 1. Mean coupling time

References

1. Bremaud, P. 1993. Maximal coupling and rare perturbation sensitivity analysis. Manuscript.

2. Chung, K.L. 1954. On a stochastic approximation method. *Annals of Mathematical Statistics* 25 : 463-483.

3. Dai, L., and Y.C. Ho. 1994. Structural infinitesimal perturbation analysis (SIPA) for derivative estimation of discrete event dynamic systems. *IEEE Trans. on Automatic Control.* To appear.

4. Doeblin, W. 1938. Exposé de la theorie des chaînes simples constantes de Markov à un nombre fini d'états, *Rev. Math. Union Interbalkanique* 2 : 77-105.

5. Doob, J.L. 1953. *Stochastic Processes.* John Wiley and Sons, New York.

6. Fu, M.C., and J.Q. Hu. 1994. Smoothed perturbation analysis derivative estimation for Markov chains. *Operations Research Letters.* To appear.

7. Fu, M.C., J.Q. Hu, and L. Shi. 1993. Likelihood ratio methods via conditional Monte Carlo and splitting. *Proc. 31st Annual Allerton Conf. on Commun., Control, and Computing* : 958-967.

8. Goldstein, S. 1979. Maximal coupling. *Z. Wahrsch. Verw. Gebiete* 46 : 193-204.

9. Hoeffding, W. 1940. Masstabinvariante Korrelationstheorie. *Schriften des Mathematischen Instituts und des Instituts für Angewandte Mathematik der Universität Berlin* 5 : 179-233.

10. Iosifescu, M. 1980. *Finite Markov Processes and Their Applications.* John Wiley & Sons, New York.

11. Lindvall, T. 1992. *Lectures on the Coupling Method*, John Wiley & Sons, New York.

Stereo Vision via Markov Models

Davi Geiger[1]
Courant Institute, NYU
251 Mercer Street
New York, NY 10012

Laurence T. Maloney
Psychology, NYU
6 Washington Place
New York, NY 10003

Krishna Kumaran
Department of Physics
Rutgers University
Piscataway, NJ 0885

Abstract

Binocular stereo is the process of obtaining depth information from two simultaneous views of a scene. The performance of computational stereo algorithms remains distinctly inferior to human performance, and at the same time, many phenomena in human stereo vision (transparency, Da Vinci stereopsis, subjective contours) have no natural place in these algorithms.

Through the framework of hidden Markov models we formulate a stereo algorithm that match along epipolar lines (one dimensional). Our aim is to introduce, in a simple and yet general manner, constraints that can account for these psychophysical phenomena. The first result is a distinctive neighborhood structure for our model and these various properties of stereo, such as occlusions, disparity discontinuity, transparency, are accounted. In order for us to fully address the problem of subjective contours the model is extended to two dimensions and a Markov random field model emerges. Finally we discuss the primitives of stereo, proposing differential geometric features, and studying their relation to occlusions, discontinuities and transparency. Initial experiments demonstrate the power of our formulation.[2]

1 Introduction and background

A biological visual system or computer vision system that uses binocular stereo transforms the disparities between features in the two images into information about the shapes and locations of objects in the scene. In the 1960's and later, experience with the Julesz random dot stereogram led to the development of several computational theories of disparity processing in biological vision including those of Sperling[19], Julesz [9]; Marr and Poggio[11] [12]; Pollard, Mayhew and Frisby[18]; Grimson[8]; Baker and Binford [2]; Kanade and Okutomi[10]. The performance of binocular stereo algorithms, including those cited above, has proven to be markedly inferior to that of biological vision systems.

These algorithms also neglect sources of stereo information corresponding to transparency, occlusion, and subjective contour phenomena in human vision. In many scenes, there are portions of the scene visible to one camera/eye that are not visible to the other (half-occlusion regions). The experiments of Nakayama and Shimojo

[2]Part of the work done when visiting the Isaac Newton Institute of Mathematics, Cambridge, UK.

[15] suggest that these regions play an influential role in human binocular stereo vision. Geiger, Ladendorf and Yuille [7] and independently Belhumeur and Mumford [3] have recently proposed models that address the relation between discontinuities and half-occlusions (termed by [3]). Much workremains to be done in this area. The phenomenon of transparency is marginalized by typical computational theories of binocular stereo, even though disparity cues alone can give the human observer a strong impression of transparency[16]. Interactions among epipolar lines are commonly discarded or underused. Indeed, there are no specific performance ciriteria proposed for such interactions (other than "looks good"). Epipolar lines are corresponding lines in the left and right images.

1.1 Our approach

Our main intents are to develop a robust, efficient, and simple stereo algorithm that outperforms existing algorithms and to provide alternative explanations for a variety of recent psychophysical experiments and suggest new experiments. We are developing a theoretical framework for binocular stereo in which occlusion, transparency, and subjective contours are treated in a unified way (all are aspects of the same problem). It is a Bayesian approach starting with differential geometric primitives designed to detection the kinds of discontinuities that signal occlusion and transparency phenomena. These features are paired bipartite windows (*D-windows*). The hidden Markov model (which includes prior information) guides the reconstruction of a surface that best matches these primitives. In the hidden Markov model, our observations are the features extracted from the stereo pairs. The states are given by the possible disparity values and the transitions are guided by the prior surface model we impose, piecewise smooth surfaces with constraints on transparency and occlusions. When interactions among epipolar lines are included then a Markov random field model naturally emerges.

Our presentation can be sketched as follows:

1. Stereo primitives ('D-windows') and candidate correspondences.
2. Occlusion, transparency, and subjective contours via a hidden Markov model.
    a. Half-occlusions to discontinuities. We start building the hidden Markov model for surface reconstruction considering our previous work that relates half-occlusions to discontinuities.
    b. Transparency. We discuss criteria to complete (fill in) the disparity information on half-occluded (unmatched) areas, taking into account Metelli's rule [13], and how it may affect the full recovery of depth.
    c. Subjective contours. We complete the model by introducing epipolar line interactions and generalizing it to a two dimensional Markov random field model. We argue that appropriate modeling (we are here defining performance criteria) of the epipolar line interactions, also known as figural continuity, yields the formation of subjective contours.
3. Initial experimental results targeted to 2.

2 Bayesian Reconstruction: A hidden Markov model

The Bayesian approach assumes that we can express the probability of the scene S given input data I by a distribution $P(S|I)$ which, by Bayes' theorem, can be written as $P(I|S)P(S)/Z$, in terms of the *imaging model* $P(I|S)$, the prior model $P(S)$ and a normalization constant Z. In addition, we can impose *hard constraints* on the possible solutions S^*. We assume that the optimal interpretation S^* is obtained by the *maximum a posteriori* estimate, $S^* = ARG\{MAX_S\}P(S|I)$. The theory can be described either in terms of solving a matching problem or in terms of surface reconstruction. It gives a compromise between accurate fitting of the data and conformance to a prior model of surfaces.

2.1 Observations or input probabilities

For each candidate match where a feature at l, from the left eye, matches a feature at r, from the right eye a disparity $d = r - l$ is considered. In the matching space this information corresponds to paving the space with data accounting for the strength of the match. The matching space is a two-dimensional space where the axes are given by the epipolar lines of the left and right images and each element of the space, $M_{l,r}$, decides whether a feature at pixel l, in the left image, matches a feature at pixel r, in the right images (see figure 1). A solution for the matching problem, a disparity map, along an epipolar line is represented by a path in the matching space.

Primitives: D-windows. The stereo primitives we employ are small windows in each of the two stereo images. The windows are halves of elliptical disks (ellipses), shaped like a filled-in 'D' and we will term them *D-windows*. A D-window is specified by giving the center of its elliptical boundary, the two axes , and the angle (zero to 2π) of the larger axis (bisecting the D-window). For any D-window there is a *complementary D-window* that is the other half of the elliptical disk (Figure ??b). D-windows are transformed into D-windows under affine (linear) transformation. . We wish to find those affine correspondences that represent 'good' matches between the contents of the D-window in the left image (under affine transformation) and the corresponding D-window in the right image. For each "good" affine of D-windows correspondence there is a complementary D-window measure, providing two sets of parameters. It is often the case that the parameters are the same. At certain points, there is no acceptable affine mapping of the complementary windows. This suggests there is a half-occlusion region. The last possibility is that there is an affine mapping of the two complementary windows, but the set of affine parameters is markedly different. This suggests that the orientation of the surface in the scene has changed discontinuously, but that no occlusion has occurred. This information will also prove useful in reconstructing the three-dimensional structure of the scene.

As in [11] [7], we use a matching process $M_{l,r}$ that is 1 if, a feature at pixel l in the left eye matches a feature at pixel r in the right eye, and is 0 otherwise. We define the probability of observing a pair of inputs, $D^{L,\pi}$ and $D^{R,\pi'}$, given the matching process M, by

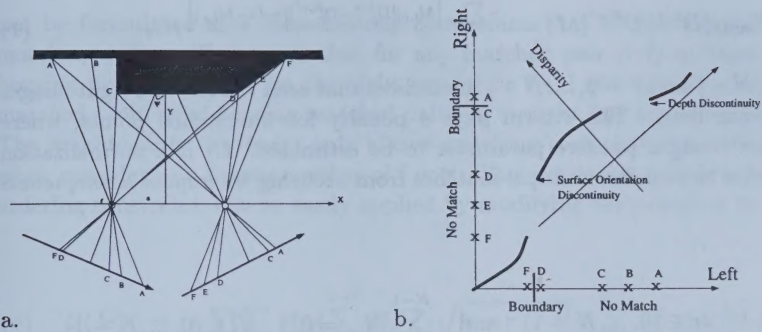


Figure 1: *(a) A polyhedron (shaded area) with self occluding regions and with a discontinuity in the surface-orientation at feature D and a depth discontinuity at feature C. (b) A matching space for the image of the ramp above. It has elements $M_{l,r}$ that decide if a feature at pixel l, in the left epipolar line, matches to a feature at pixel r, in the right epipolar line.*

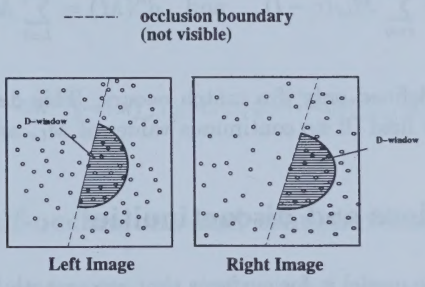


Figure 1a

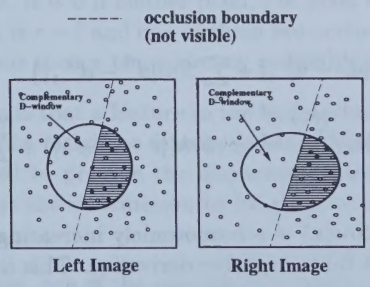


Figure 1b

Figure 2: D-windows and its transformations. a. exhibits an D-window that is the image of the D-window on the left. They are, of course, half-elliptical in shape, and the other half of the ellipse is the *complementary D-window* (b).

$$P_{input}(D^{L,\pi}, D^{R,\pi'}|M) = e^{-\sum_{l,r}\left[M_{l,r}\|D_l^{L,\pi} - D_r^{R,\pi'}\| + \epsilon(1-M_{l,r})\right]}/C_1 \quad (1)$$

where $l = 0, ..., N-1$ and $r = 0, ..., N-1$ are indices that scan the left and right images along the epipolar lines. The ϵ term pays a penalty for unmatched points, where $M_{l,r} = 0$, with ϵ being a positive parameter to be estimated. C_1 is a normalization constant. In order to prohibit multiple matches from occuring we impose a *uniqueness* constraint:

$$\sum_{l=0}^{N-1} M_{l,r} = 0,1 \quad \forall r \in (0, ..., N-1) \quad \text{and} \quad \sum_{r=0}^{N-1} M_{l,r} = 0,1 \quad \forall l \in (0, ..., N-1). \quad (2)$$

Notice that this uniqueness guarantees that there is at most one match per feature, but also permits unmatched features to exist. We define a disparity field for the left eye, d^L, and another for the right eye, d^R, as

$$d_l^L(M) = \sum_{r=0}^{N-1} M_{l,r}(r-l), \quad \text{and} \quad d_r^R(M) = \sum_{l=0}^{N-1} M_{l,r}(r-l).$$

where d^L and d^R are defined only if a match occurs. This definition leads to integer values for the disparity field (if no continuous values of $M_{l,r}$ are considered).

2.2 Half-occlusions and discontinuities

The initial prior surface model is for surfaces that are smooth but also allows for large diparity changes to occur without high penalties. We suggest to use the potential

$$\begin{aligned} U(d^L, d^R(M)) &= \mu[\sum_l Sm|d_l^L - d_{l-1}^L| + \sum_r Sm|d_r^R - d_{r-1}^R|] \\ &= \mu \sum_l \sum_r M_{r,l}[\sum_{r'=0}^{N-1} M_{r',l-1} Sm(|r-r'-1|) + \sum_{l'=0}^{N-1} M_{r-1,l'} Sm(|l-l'-1|)] \end{aligned}$$

where μ is a constant, and $Sm(x)$ is a monotonicly increasing smooth function, such that after some value of x, it takes negative derivative. This implies that the function increases less and less as x increases. The function $\sqrt{(x)}$ is a good candidate for prior and we adopt in this paper. We emphasize that the choice of prior is motivated by the class of stimuli on which the system is designed to work. The occlusion constraint [7].

Proposition 1 (Occlusion constraint) *A discontinuity in one eye, along the epipolar line, corresponds to an occlusion in the other eye. Moreover, the prior cost for an occlusion in the left and right eye must be the same.*

can be formulated as a *monotonicity constraint.* Plot all the matched pairs in the matching space. We require that for any matched pair (r, l) at least one of neighbouring points at $r + 1$ (in the right image) or $l + 1$ (in the left eye) must also be matched. Join neighbouring matched pairs by straight line segments to form a curve. The *monotonicity constraint* only allows matching such that this curve is monotonic when consider as a function either of l or r. This constraint, which is a variant of the ordering constraint, can be easily applied by modifying the potential to

$$U(M) = \mu \sum_l \sum_r M_{r,l} \left(\sum_{r'=0}^{r-1} M_{r',l-1} \sqrt{|r - r' - 1|} + \sum_{l'=0}^{l-1} M_{r-1,l'} \sqrt{|l - l' - 1|} \right)$$

and assigning a Gibbs probability distribution for the prior and combining it with (1), using Bayes' theorem, we obtain

$$P_{stereo}(M| \parallel D_l^{L,\pi} - D_r^{R,\pi'} \parallel) = \frac{e^{-\epsilon N^2}}{Z}$$

$$\prod_{r=0}^{N-1} \prod_{l=0}^{N-1} e^{-M_{l,r} \left[\|D_l^{L,\pi} - D_r^{R,\pi'}\| - \epsilon + \mu \left(\sum_{r'<r} M_{r',l-1} \sqrt{|r-r'-1|} + \sum_{l'<l} M_{r-1,l'} \sqrt{|l-l'-1|} \right) \right]} \quad (3)$$

2.3 Filling in half-occlusions and transparency

Filling in the disparity (surface) for the half-occluded regions may affect the overall solution of the correpsondence. We introduce a new matching-occlusion-transparency field, T_{lr}, which is 1 or 0. It is 0 if neither pixel, l or pixel r, is occluded. It is 1 only if the disparity estimation is $r - l$ and one of them are occluded thus, we have never the case $T_{lr} = M_{lr} = 1$. In our theory transparency can only occur when $T_{lr} = 1$.

The first observation is that a feature in the half-occluded area is likely to have the same disparity as a similar neighbor feature (eventually one neighbor feature nearby will not be half-occluded to provide the disparity information). The neighbor with larger similarity provides the information to be given to the occluded pixels. This is the fill in constraint.

Transparency: Always, one of the possible solutions considers the occluded surface to be in front of the matched one ($M_{lr} = 1$). Then, a phenomena of transparency must occur or else how come occluded features are perceived in front ? In these cases, we must consider transparent requirements and we recall Metelli's transparent rules [13] to include in the energy function. Thus, $Met^L(l, r, l', l'')$ is 1 if $I^L_{l''}$ could be in front of $I^L_{l''+l'-l+1}$ such that $I^R_{r-l+l''}$ would be seen in the right eyeand is zero otherwise. Here, $l - l'$ is the size of the half-occluded region (i.e. $M_{lr} = 1$ and $M_{l',r-1} = 1$) and l" is some pixel in the occluded region. Analogously $Met^R(r, l, r', r'')$ is 1 or zero, in the right image.

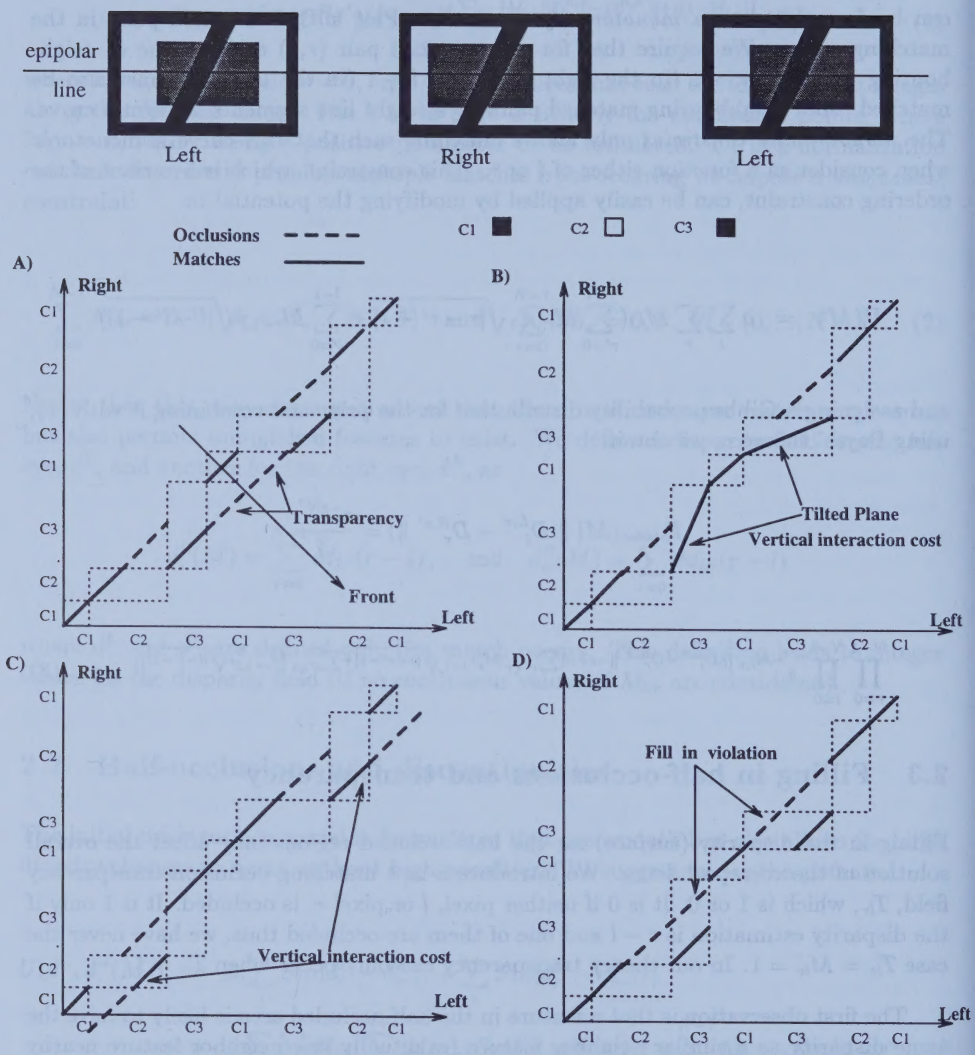


Figure 3: *Transparency from stereo (After Nakayama and Shimojo, 1992). We analyse four situations. Notice that the solution at top (and bottom) epipolar lines gives zero disparity everywhere. The dotted boxes represent regions of possible matches (acceptable similarity measures). A. Actual solution. The transparency is a consequence of the filling in process. Note that the filling in process has limits of diffusion that can be tested via psychophysical experiments. B. The tilted plane solution, possible along epipolar lines, but with high cost when vertical interactions are added since, different epipolar lines will give different tilted angles (if the black inclined bar were vertical the tilted solution would be also plausible). C. Again a possible epipolar solution (with the same cost as A.), but becomes costly when vertical interactions are added (because of region C2, white). D. Not a possible solution due to the fill in constraint (fill in with similar neighbors disparity).*

In the case where transparency is not possible a third surface possibility of a tilted plane must be included (See figure 3 B.). We propose to add the energy term

$$U^{T,L}_{r,l,l'} = \sum_{l''=l'+1}^{l-1} \Big\{ T_{l'',r-1+l''-l'} \parallel D^{L,\pi}_{l''} - D^{L,\pi'}_{l'} \parallel + [Met^L(l,r,l',l'')\, T_{l'',r+l''-l} + T_{l'',r}] \parallel D^{L,\pi}_{l''} - D^{L,\pi'}_{l} \parallel \Big\}$$

$$U^{T,R}_{r,l,r'} = \sum_{r''=r'+1}^{r-1} \Big\{ T_{l-1+r''-r',r''} \parallel D^{R,\pi}_{r'} - D^{R,\pi'}_{r''} \parallel + [Met^R(r,l,r',r'')\, T_{l+r''-r,r''} + T_{l,r''}] \parallel D^{R,\pi}_{r} - D^{R,\pi'}_{r''} \parallel \Big\}.$$

Thus, three cases are considered for only one solution: (i) The first one gives a regular occlusion (ii) gives a transparent "occlusion" and (iii) gives a tilted surface (see figure 3). An improvement over this cost for the disparity estimation at occluded regions is via the $D - windows$ information since, their parameters provide information about the surface. We plan to explore this possibility theoretically and experimentally.

We can use a dynamic programming algorithm [4] to solve for the optimal path due to the local neighbor interactions. One of the first works on stereo using dynamic programming was by Baker and Binford [2] and later ([17]).

2.4 Epipolar line interactions and subjective contours

Epipolar line interaction play an important role on the formation of a depth map from a stereo pair. Many theories of stereo address this issue. We propose that epipolar interaction's models to be measured on their capacity of predicting subjective surfaces/contours (true positive case). Moreover some transparent phenomena are fully understood (because of disambiguation) when epipolar line interaction take place.

We observe that subjective contours are always perceived with a perception of a surface in front of another one (occluding another one). The stereo process yields the perception of depth surfaces and thus, cause strong subjective contours sensations. In the example of the stereo Kanizsa triangle (see figure 4), there are epipolar lines with zero disparity, e.g. in the middle of the figure. We conjecture that the depth perception is consequence of the epipolar line interaction that enforces piecewise smooth depth maps. We are considering adding a vertical interaction energy term of the form

$$U^{ep} = \sum_e \Big\{ \sum_r (1 - \gamma eh^R_r) \sqrt{d^{R,e}_r - d^{R,e-1}_r} + \sum_l (1 - \gamma eh^L_l) \sqrt{d^{L,e}_l - d^{L,e-1}_l} \Big\},$$

where eh^R_r and eh^L_l are the intensity edge maps of the right and left image, i.e. they are 1 if there is an edge and zero otherwise, γ is an attenuation factor smaller (or

equal) to 1. This term will effectively block the epipolar interactions when an intensity edge separate the epipolar lines. The model becomes a Markov random field, in two dimensions, and we can no longer use dynamic programming (unless approximations between these interactions are made, e.g. mean field approximation).

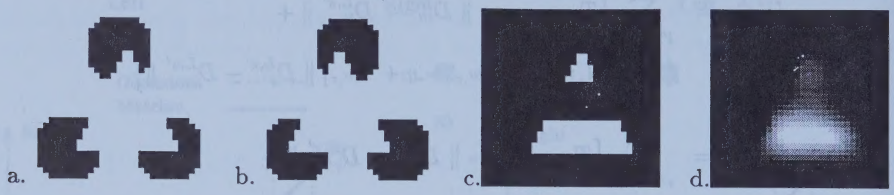


Figure 4: *(a) and (b) 32 by 32 pixels of the Left and right Kanisza triangle. (c) solution along epipolar lines (using a bias term twoards zero disparity) (d) reconstruction of disparity from sparse data (assuming information availble from (c) just at intensity edges.*

2.5 Pilot experiment

The algorithm for (3), without using the filling in process and the epipolar line interaction, has already been tested on real images with positive results (see [7]). We have so far conducted pilot experiments with black and white images to test these increments. The features are single pixels. However, along vertical interactions we have implemented a simpler approach. We simply run the algorithm along epipolar lines and selected the information at intensity edges (where is more reliable). Then a reconstruction of 2D, from these sparse data (disparity data at intensity edges) provides a full recover of subjective surfaces. We could alternatively use the hidden markov model machinery to estimate the confidence for each estimate, but further analysis needs to be done. Figure 4 is a typical pilot experiment using the Kanizsa triangle and the results are supportive of the model.

Acknowledgements: D.G. would like to thank the hospitality and environment of the Isaac Newton Institute. In particular, discussions with A. Blake and D. Mumford and J. Mayhew were extremely helpful.

References

[1] N. Ayache. *Artificial Vision for Mobile Robots* The MIT Press. Cambridge, Mass 02142. 1991.

[2] H.H. Baker and T. O. Binford. Depth from edge and intensity based stereo. In *Proceedings IJCAI*, pages 631–636, Vancouver, 1981.

[3] P.N. Belhumeur and D. Mumford. A bayesian treatment of the stereo correspondence problem using half-occluded regions. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, 1992.

[4] R. E. Bellman. *Dynamic Programming.* Princeton University Press, 1957.

[5] A. Blake and A. Zisserman. *Visual Reconstruction.* MIT Press, Cambridge, Mass, 1987.

[6] D. Buckley and J. Frisby and J. Mayhew. Integration of stereo and texture cues in the formation of discontinuities during three-dimensional surface interpolation. *Perception*, 18-5:563–588, 1989.

[7] D. Geiger and B. Ladendorf and A. Yuille". Binocular stereo with occlusion. *2nd ECCV-Lecture Notes on Computer Science*, Springer-Verlag, May 1992.

[8] W. E. L. Grimson. *From Images to Surfaces.* MIT Press, Cambridge, Mass., 1981.

[9] B. Julesz. *Foundations of Cyclopean Perception.* The University of Chicago Press, Chicago, 1971.

[10] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: theory and experiments. In *Proc. Image Understanding Workshop DARPA*, PA, September 1990.

[11] D. Marr and T. Poggio. Cooperative computation of stereo disparity. *Science*, 194:283–287, 1976.

[12] D. Marr and T. Poggio. A computational theory of human stereo vision. *Proceedings of the Royal Society of London B*, 204:301–328, 1979.

[13] F. Metelli. The perception of transparency. *Scientific American*, 230 (90):91-98, April 1974.

[14] D. Mumford and J. Shah. Boundary detection by minimizing functionals, i. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, San Francisco, CA, 1985.

[15] K. Nakayama and S. Shimojo. Da vinci stereopsis: depth and subjective occluding contours from unpaired image points. *Vision Research*, 30:1811–1825, 1990.

[16] K. Nakayama and S. Shimojo". Experiencing and Perceiving Visual Surfaces. *Science*, 257:1357–1363, 1992.

[17] Y. Ohta and T. Kanade. Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7(2):139–154, 1985.

[18] S. B. Pollard, J. E. W. Mayhew, and J. P. Frisby. Disparity gradients and stereo correspondences. *Perception*, 1987.

[19] G. Sperling. Binocular vision: A physical and a neural theory. *American Journal of Psychology*, 83:461–534, 19670.

On the Applications of Morphological Methods for Dynamic Object recognition *

Prabir Bhattacharya and **Weibin Zhu**
Department of Computer Science & Engineering
University of Nebraska-Lincoln
Lincoln, NE 68588-0115, USA

Abstract

We develop a new morphological hit-miss algorithm for detecting the features of binary images. The standard hit-miss algorithms use block structure patterns for matching features but in many practical situations this may cause problems because the size of the object is usually unknown. In the proposed algorithm we match the corners of the object instead of the block structure pattern – this approach works nicely because most objects have different structure elements for the corners.

1 Introduction

The morphological hit-miss transformation is a basic tool for the shape detection [10]. The main idea of this method involves the erosion of an image X with a foreground structuring element B, and the erosion of the complemented image X^c with a background structuring element D, the intersection of these two erosions is the *hit-miss transform*. The *hit-miss algorithm* uses *erosion* and *dilation* operations [10]. The definition for the *erosion* of an image X with a structuring element B is:

$$X \ominus B = \{a : B_a \subseteq X\} \tag{1}$$

where $\ominus$ denotes the *erosion*, B_a is a set B shifted by a, and $\subseteq$ denotes a subset. The definition for the *dilation* of an image X with a structuring element B is:

$$X \oplus B = \{a : B_a \cap X \neq 0\} \tag{2}$$

where $\oplus$ denotes dilation and $\cap$ is the intersection (i.e., AND operation). The *hit-miss transformation* is defined as:

$$X(B, D) = [(X \ominus B) \cap (X^c \ominus D)] \tag{3}$$

where we denote a binary image by X, its complement by X^c, a foreground structuring element by D. The traditional algorithm is necessary for each object to maintain a large database containing templates of varied sizes. Since the size of a target (signature) changes, in general, from image to image, we cannot always apply this algorithm efficiently.

Recently, Casasent *et al.* [2] developed an optical implementation of the morphological hit-or-miss transform, improving the standard optical pattern-recognition correlator techniques for the

*Support by the US Air Force Office of Scientific Research under Grant AFOSR F49620-94-1-0029 is gratefully acknowledged.

detection of targetsin images containing clutter. They modified the standard hit-or-miss transformation to do rank-order filtering where the foreground structuring element B can be the desired object, and the background structuring element D can be its complement with a border (background) around it. To choose B and D, according to their algorithm, it is not necessary to select $D = B^c$ — the erosion operation can be performed on an optical correlator with output correlation threshold chosen to select the desired rank-order filter. The advantage of their approach over the standard hit-or-miss transform is that it gives some flexibility in choosing the size of the templates. However, this approach still needs some pre-processing to get the approximate size of the target, and then to choose a set of structure elements whose sizes are close to the size of the desired object. Thus, it is necessary for each object to maintain a large database containing templates of varied sizes. Since the size of a target (signature) changes, in general, from image to image, we cannot always apply this algorithm efficiently.

In this paper, we propose a polynomial hit-miss algorithm which does not need any pre-processing to compute the size of the target. The main idea of our proposed algorithm is to recognize the corners of an object rather than the whole object itself. By recognizing the corners of the object, we can recognize the shape of object itself. A novel feature of our algorithm is that the size of the object is relatively unimportant in the recognition process. Our proposed algorithm can be implemented in many different case for shape detection. Also, it can be applied when we have a set of models and we wish to match the unknown object with the given models – the recognition process would be to calculate the highest percentage of match. For the templates for each model in the manner above, and apply the hit-miss transform on the unknown image for the templates of each model. The proposed algorithm works nicely particularly with objects which have polyhedral shapes; for curved surfaces, the algorithm can detect the corners (if any). Thus for a box-like figure (e.g., a tank or a gun), the algorithm can be applied successfully. Since the templates designed to recognize a certain shape signature, and same templates cannot be applied to match other shapes. So the algorithm is expected to recognize the shape correctly. We have given the implementation of the proposed algorithm for the recognition of the muzzle of a tank gun.

To represent the templates and binary images conveniently, we shall use a polynomial representation. This algebraic methodology has been used by us extensively in past to process binary, gray, colored, and 3-D voxel-based images [1], [6]-[9]. In the polynomial approach, a binary image of size $r \times s$ is represented by a polynomial in two variables. Similar algebraic representation is used for the templates. Various image processing operations have been defined in terms of certain algebraic operations on these image and template polynomials. The morphological operations dilation and erosion can be performed conveniently using polynomial approach, the details are given in [1] and [8]. Since the hit-miss transform uses dilation and erosion as the basic operations, we can implement the hit-miss transform using the algebraic approach. The possible advantage of the polynomial approach lies in the fact that if we use the *Fast Fourier Transform* (FFT) to computer the convolutions of polynomials involved in the processing, a considerable speed-up is obtained (this is explained in detail in [1] and [9] where estimates of the speedup have been given). In the polynomial approach, a binary image of size $r \times s$ is represented by a polynomial in two variables:

$$\sum_{ij} a_{ij} X^i Y^j$$

where the summation is over all the pixels of the image and a_{ij} is 1 or 0 according as the pixel is black or white respectively. Similar algebraic representation is also used for the templates. The morphological operations of dilation and erosion can be performed using polynomial approach, the details are given in [1] and [8]. We shall review it briefly in the following paragraph. Let A and B be the picture polynomials of an binary image and a structuring element (SE) respectively. Let

$$P = A * B \equiv \sum_{ij} p_{ij} X^i Y^k$$

denote the usual polynomial product of A and B. Then the dilation $\oplus$ and erosion $\ominus$ are obtained as follows:

$$A \oplus B = \sum_{ij} c_{ij} X^i Y^j, \quad A \ominus B = \sum_{ij} d_{i,j} X^i Y^j$$

where $c_{ij} = 1$ if $p_{ij} \geq 1$ and 0 otherwise, and $d_{ij} = 1$ if $p_{ij} = n$ and 0 otherewise where n is the number of nonzero terms in the polynomial B. Since the primitive operations of hit-or-miss transformation are dilation and erosion, it follows that we can use the polynomial representation for the hit-or-miss transformation.

2 Method

We shall match the corners of the unknown image with a set of templates. Each of the corners corresponds to two templates, one for the foreground, and the other for the background. In the foreground template-matrix, we attempt to capture the features of the corner pixel, by mapping it and its close range pixels to the entries of the template-matrix with corresponding values; in the background template-matrix, we do the same thing to capture the feature of complement of corner pixel and its nearby pixels.

Now we can apply the standard hit-miss operation using this set or templates to match each corner separately. The foreground templates are used for the hit operation and background templates are for the miss operation. Since we are matching the corners, the size of the unknown object is irrelevant,we can actually calculate the size of the object after we match the corners.

Suppose we have an object with n corners, for each corner we choose a pair of templates B_i, D_i, where B_i denotes the foreground template and D_i denotes the background template $(1 \leq i \leq n)$. Using these templates the hit-miss transform given in equation (3) can be expressed as:

$$X(B_i, D_i) = [(X \ominus B_i) \cap (X^c \ominus D_i)] \quad (1 \leq i \leq n) \tag{4}$$

After implementing algebraically the hit-miss transform given by equation (4) for each corner, we calculate the polynomial:

$$X_{final} \equiv \sum_{i=1}^{n} X(B_i, D_i) \tag{5}$$

where the summation is the usual addition of polynomials. In the polynomial X_{final}, the terms corresponding to the n-highest coefficients would represent the n corners of the unknown image. We can also obtain algebraically the positions of the corners from the polynomial X_{final} – if the term $X^i Y^j$ has a n-highest coefficient, then the corresponding corner is at the (i, j) position. Knowing the positions of all the corners, we can estimate the size of the unknown object.

Our proposed algorithm also has the advantage for the real-time computation. If the object has large size, the standard hit-miss transform needs relatively large memory space to compute. On the other hand, our proposed algorithm just needs much smaller constant size templates. Furthermore,

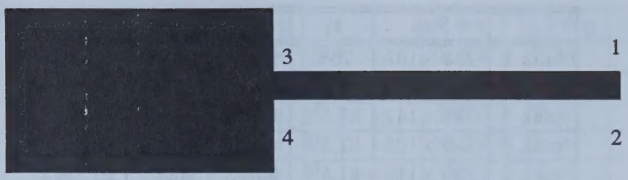


Figure 1: Shape of a tank gun turret

since our algorithm matches corners instead of the whole object, and each corner is independent of each other, it can be also implemented directly on a single-instruction-multiple data (SIMD) parallel computer,resulting in a speedup. The run-time of our proposed algorithm is, clearly, $O(mn)$, where the m and n denote the number of rows and columns, respectively, of the unknown image.

After recognizing the corners, a few clean-up steps may be done as follows. We can use a standard filter to reduce the noise of the image. Further, to determine if there is a solid object bounded by the corners, we may check the the coefficients of those terms in the polynomial representation of the image which correspond to the pixels inside the the boundary. If the corners are given, then it is easy to draw the boundary between the corners. So, we can also check whether the boundary formed by the corners, is free of block pixels. These last two steps can be redesigned for different applications.

3 Implementation

For our implementation, we choose tanks as our objects. In order to recognize a tank, it will be convenient to define the templates for the the purpose of matching of muzzles of tank guns. A tank muzzle gun has rectangular shape which is connected on one side to the turret of the tank, we choose a pair of four templates. For each corner i we define B_i which is the foreground template and D_i which is the background template ($1 \leq i \leq 4$). In the design of these templates, we try to match not the actual corner pixel but a pixel close to it corresponding to a point on the muzzle gun, we call this pixel a *pseudo-corner*. The idea behind this is to compensate for the noise in the immediate neighborhood of the corner, if we match only the corner point, then there may be a possible error due to the noise (the corner pixel may even be missing due to the blur). In Figure 1, we give the definition of the templates used by us. For each template we have marked the entry corresponding to the pseudo-corner with the symbol $\langle\rangle$.

The motivation behind the design of these templates can be seen as follows. The pairs of corners that have the same profiles for the foreground, are assigned the same foreground templates for each of these pairs, in other words, $B_1 = B_2$ and $B_3 = B_4$. However, for the array B_1, the $(1,3)$ entry corresponds to the actual corner, whereas for B_2 the $(3,3)$ entry corresponds to the actual corner. In B_3 or B_4, notice that first column corresponds to the pixels on the turret adjacent to the gun, and these are all assigned the value 1. The $(4,2)$ entry on B_3 or B_4 corresponds to actual corners. In D_1 the first row and last column represent the pixels in the background which are adjacent to the contour of the gun at the corner 1, similar motivation applies to D_2, D_3 and D_4.

The algorithm presented in the paper has been tested successfully on a Sun Sparc 2. The statistics of our implementation on six tank images are given in Table 1.

	Size	t_1	t_2	t_3	t_4
tank 1	263 ×107	75%	93.5%	80.9%	71.4%
tank 2	340 ×140	93.5%	93.5%	85.7%	76.2%
tank 3	338 ×147	87.5%	87.5%	90.6%	80.9%
tank 4	240 ×102	87.5%	100%	90.6%	90.5%
tank 5	285 ×111	81.5%	93.5%	90.6%	90.5%
tank 6	320 ×120	75.0%	75.0%	56.0%	75.0%

Table 1: Statistics on tank images

In Table 1, t_1, t_2, t_3, t_4 denote the four corners of a tank gun – in Figure 1 we have denoted these corners, for convenience, by 1–4 respectively.

Similarly, we have designed templates for the recognition of submarines. We omit the details.

3.1 Further Modifications

The results given above show that our algorithm works quite nicely. However, in many situations tanks do not have guns in the horizontal position. In order to recognize the tank guns with elevating angles, we now add some further steps to make it work more realistically.

First, we need to determine the trade off between accuracy and efficiency. If the real-time application need very accurate elevating angle, it will cost extra memory and speed for the computing. In our example, the roundoff angle is about 10 degrees which means that we set up the model for every 10 degree in general. Second, we need to rotate the real image to about 10 degrees every time. We can not simply rotate the templates, because after rotation, the dimensions of each template will change, and more importantly it will have extra blank pixels inside of templates which makes it difficult to run the hit-miss transformation. So rotating the whole image will avoid this problem. Then, apply our proposed algorithm to match each rotated image. During each match, we need check the false alarm. We eliminate the match which is not proximate rectangle shape. After all the matches, we select the best result to be our best match, which also correspond to the degree of that image has been rotated.

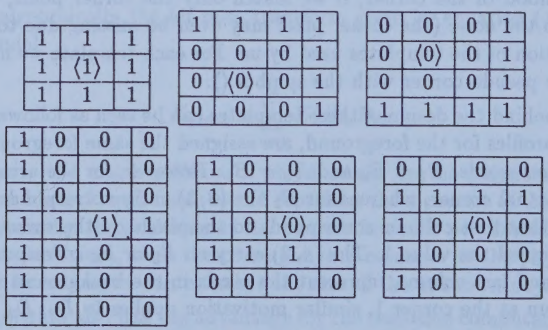


Figure 2: Templates for $B_1 = B_2$, D_1, D_2, $B_3 = B_4$, D_3 and D_4 respectively

	0 deg	11.5 deg	22.5 deg	45 deg
t_1	87.5%	70.5%	False	False
t_2	100%	81.5%	False	False
t_3	90.6%	87.5%	False	False
t_4	90.5%	81.5%	False	False
result	select	drop	drop	drop

	0 deg	11.5 deg	22.5 deg	45 deg
t_1	78.5%	False	56.6%	False
t_2	91.5%	False	76.8%	False
t_3	87.5%	False	81.5%	False
t_4	90.5%	False	45.5%	False
result	Select	drop	drop	drop

	0 deg	11.5 deg	22.5 deg	45 deg
t_1	91.5%	False	False	False
t_2	81.5%	False	False	False
t_3	100%	False	False	False
t_4	90.5%	False	False	False
result	Select	drop	drop	drop

Figure 3: False Alarm Rates

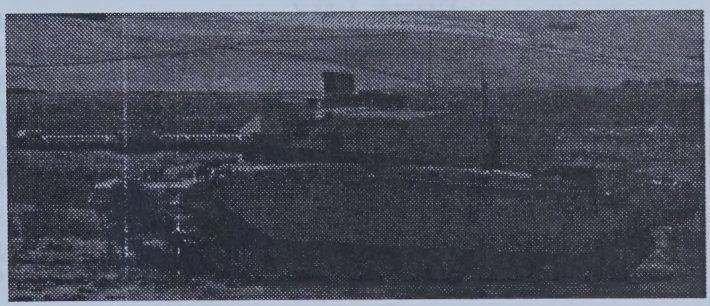


Figure 4: Tank 1

4 Conclusion

In this paper, we implement our algorithm to recognize the tank muzzle gun by matching the four corners of the rectangle. This approach gives an improvement over other standard hit-miss transform based algorithms because the size of the of the object does not play a crucial role in the proposed algorithm. Matching the corners of the object makes the size of image relatively unimportant for the recognition. We represent the images and the templates using an algebraic approach which has been used by us extensively in the past and its advantages have discussed elsewhere in detail. In a future work we plan to consider the problem of noise in the image.

References

[1] P. Bhattacharya, K. Qian and X. Lu, An Algebraic Approach to Morphological Operations on 2D and 3D images, *Patt. Recog.*, **26**. 1785-1796 (1993).

[2] D. Casasent, R. Schaefer, and R. Sturgill, Optical hit-miss morphological transform, *Applied Optics*, **31**, (no. 29), 6255-6283 (1992).

Figure 5: Tank 2

[3] D. Casasent and R. Schafer, Multilevel Hit-Miss Transform for Object Recognition, to appear in, *Proc. SPIE*, vol. 2353, Nov., 1994.

[4] E.R. Dougherty and R.P. Loce, Optimal mean–absolute–error hit–or–miss filters: morphological representation and estimation of the binary conditional expectation, *Opt. Eng.* **32**, 815-827 (1993).

[5] R.M. Haralick, S.R. Sternberg, and X. Zhuang, Image Analysis Using Mathematical Morphology, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **9**, 532-550 (1987).

[6] K. Qian and P. Bhattacharya, Binary Image Processing by Polynomial Approach, *Pattern Recognition Lett.*, **11**, 395-403 (1990).

[7] K. Qian and P. Bhattacharya, A Template Polynomial Approach for Image Processing and Visual Recognition, *Pattern Recognition*, **25**, 1505-1515 (1992).

[8] K. Qian and P. Bhattacharya, Determining Holes and Connectivity in Binary Images, *Computers and Graphics*, **16**, 283-288 (1992).

[9] K. Qian, X. Lu and P. Bhattacharya, A 3D object recognition system using voxel representation, *Pattern Recognition Lett.*, **13**, 725-733 (1992).

[10] J. Serra, *Image Analysis and Mathematical Morphology*, Academic Pr., London, 1982.

[11] R.C. Vogt, *Automatic Generation of Morphological Set Recognition Algorithms*, Springer-Verlag, New York, 1989.

Detection and Tracking of Vehicles from Video with Applications to IVHS*

Wai Ying Kan, James V. Krogmeier† and Peter C. Doerschuk
School of Electrical Engineering, Purdue University

September 29, 1994

Abstract

Accurate and wide-area estimates of vehicle velocity and traffic spatial and temporal densities will be essential components of future algorithms for freeway and arterial street control, for incident prediction and detection, and for optimization in route selection. Algorithms like these are an important part of current work in traffic management and traveler information as port of research and development of Intelligent Vehicle-Highway Systems (IVHS). This paper presents an approach to a class of vehicle monitoring problems which is based upon a video backbone sensor and multiple target tracking (MTT). The method allows easy integration of measurements made from other sensors; for example, inductive loops, microwave radars, laser range profilers, etc.

1 Introduction

The Intelligent Vehicle–Highway System (IVHS) [9] is envisioned to be an integrated group of technologies and services providing fundamental improvement in the efficiency and safety of the nation's surface transportation infrastructure. Central to the achievement of these goals is a traffic management system which will be required to process distributed traffic observations for the purposes of state estimation and prediction. The results of estimation and prediction algorithms are used for traveler information systems, traffic control (via ramp metering [6] and signal coordination), and incident response.

In this paper we describe our application of video image processing and multiple target tracking (MTT) to the problem of traffic monitoring. The basic idea is to use low frame-rate video to detect cars and then drive a MTT with the detections in order to estimate vehicle tracks. Video-based tracking for non-IVHS purposes is discussed in Refs. [1, 5, 8]. From the tracks it is then straightforward to compute the basic traffic parameters such as flow, speed, and concentration. If accurate tracks can be maintained over distance, then vehicle tracks can provide additional information, such as origin-destination pairs of interest to traffic planners.

*This work was partially supported by a graduate fellowship form the Purdue Research Foundation.

†Mailing address: James V. Krogmeier, 1285 Electrical Engineering Building, Purdue University, West Lafayette, IN 47907–1285; Telephone: (317) 494–3530; Fax: (317) 494–6440; Internet: jvk@ecn.purdue.edu.

Current technology based on loops [4], which are inductive coils buried in the road surface, provides presence-of-vehicle information at a single point. The loop is not portable, and in fact requires road excavation and lane closure for installation, and is failure prone in northern winters. It is, however, relatively inexpensive ($\$10^3$). Information from closely-spaced loops is combined to give local speed estimates. This and other traffic parameters are then extrapolated to larger regions.

Video image processing will play a key role in IVHS sensor technology [7, 4] because it provides wide area monitoring, it is passive and portable, and does not require the installation of special equipment in the vehicle. Currently available systems for video sensing of traffic conditions are essentially video replacements for inductive loop technology [7] in which vehicle detection is done in discrete boxes delineated in the image. The output of such a sensor consists only of vehicle detections. Vehicle velocity is obtained by placing two detectors in close proximity and measuring the time between detections.

Tracking algorithms can also play an important role in future traffic monitoring systems. MTT provides a principled methodology for the incorporation of multiple measurements from a variety of different sensors (Doppler radar, video, passive and active infrared). Incorporation of vehicle dynamics and human factors (Car Following theories [2]) allows the classical Bayesian tradeoff between observations and model and can lead, with good modeling, to vastly improved estimates of distance headway and velocity. Moreover, track formation and data association algorithms provide a measure of immunity to false alarms which is important if the system is to operate under a variety of conditions, inclement weather, darkness, etc. The tracks themselves are an interesting, though unexplored, traffic parameter. For example, lane change activity per unit time or distance could be very useful in incident detection and prediction. Quantities such as transit time come directly from tracking and not from snapshots of position and velocity.

In the remainder of this paper we describe vehicle detection based on video (Section 2), modeling of vehicle dynamics and road geometry for MTT (Section 3), the MTT algorithm (Section 4), and experimental results on real highway video data (Section 5) Due to space limitations we provide greater detail for the detector than for the MTT.

2 Vehicle Detection

The automatic detection of vehicles from individual video frames is a relatively difficult task. A robust detector must account for variable ambient lighting conditions, the diversity of vehicles to be detected, the scene geometry and its interaction with spatial sampling, camera platform vibrations, and nonlinear (optical) and nonstationary (automatic gain control) behavior of the imaging process. The problem is made particularly difficult due to the passive nature of the video sensor and the relatively complicated target geometry which rules out reasonable deterministic models for vehicle signature in the image. In addition, the positions of vehicles in their lanes and even their number are usually unknown.

In order to avoid complicated vehicle signature models we have chosen a simple statistical model for the vehicles and the background in the image. One of the central goals of this work is efficient computational realization of the entire algorithm, including detection and tracking. Therefore, we restrict the detection problem to the individual lanes themselves (call these the lane images), we use only single frames for detection, and where feasible we avoid pixel or region based image processing. Details will follow. The primary goal of the detection work is the development of detection thresholds computed

from estimated lane image statistics which will automatically adapt to changing ambient illumination. A secondary goal is good approximate expressions for vehicle detection probability and false alarm rate which may be used in the development of subsequent tracking algorithms.

2.1 Background Estimation

Examination of histograms of pixel intensities in typical lane images suggests that a three component Gaussian mixture may be a reasonable statistical model for a lane including background and vehicles. Preliminary attempts to fit such a model using the Expectation Maximization algorithm showed that a good fit could be obtained using a common variance for the three terms, a low mean to represent shadows under and alongside vehicles, a high mean to represent bright regions associated with vehicles, and an intermediate mean representing the road surface. The mixing proportions depend upon the number, size, and position of vehicles in the lane. In addition, the image sequences examined in this project have exhibited a considerable longitudinal and lateral variation in the apparent mean of background pixel intensity. These variations, due to the changing viewing angle in the longitudinal direction and to surface irregularities in the lateral direction, are modeled by assuming that the pixels associated with the road surface in the lane are independent Gaussian random variables with a common variance σ_r^2 and spatially dependent mean μ_{ij}.

For these reasons it is advisable to form a pixel by pixel estimate of the lane background, and, in addition, to allow the background estimate to slowly forget the past in order that temporal changes in illumination can be tracked. Let $\phi_{ij}(t)$ be the lane image at frame t and pixel location ij for a particular lane. We consider (ignoring vehicles for the moment) an estimator defined recursively by $\bar{\phi}_{ij}(t+1) = (1-\lambda)\phi_{ij}(t) + \lambda\bar{\phi}_{ij}(t)$ with $0 < \lambda < 1$ and initial condition $\bar{\phi}_{ij}(0) = \phi_{ij}(0)$. It is easy to show that if the $\{\phi_{ij}(t) : t \geq 0\}$ are iid Gaussian random variables with mean μ_{ij} and variance σ_r^2 that this estimator is unbiased and

$$\mathrm{Var}\{\bar{\phi}_{ij}(t)\} = \sigma_r^2 \left[\frac{1-\lambda+2\lambda^{2t-1}}{1+\lambda}\right]$$

which, in the limit as $t \to \infty$ gives a variance of $\sigma_r^2(1-\lambda)/(1+\lambda)$. Slow forgetting ($\lambda \approx 1$) clearly results in improved performance but at the cost of limiting the ability to track temporal variations in mean. Before applying the image to the detection algorithm described below we form the difference image

$$\tilde{\phi}_{ij}(t) = \phi_{ij}(t) - \bar{\phi}_{ij}(t). \tag{1}$$

This procedure removes the need to consider background mean variation in detection allowing an essentially uniform detection algorithm to be applied. In theory $\tilde{\phi}_{ij}(t)$ has mean zero and, in the limit of large t, it has variance $2\sigma_r^2/(1+\lambda)$. Next we must consider the problem of performing this background estimate in the presence of vehicles. This is accomplished by feeding back the results of the detection algorithm. Define $M_{ij}(t) = 1$ if a vehicle is detected at pixel ij at time t and $= 0$ otherwise and define the new background estimator to be

$$\bar{\phi}_{ij}(t+1) = M_{ij}(t)\left[(1-\lambda)\phi_{ij}(t) + \lambda\bar{\phi}_{ij}(t)\right] + (1-M_{ij}(t))\bar{\phi}_{ij}(t). \tag{2}$$

The difference image is still computed via (1).

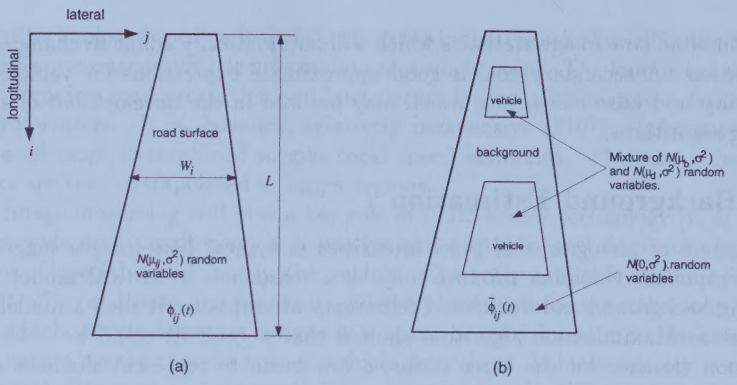


Figure 1: Illustration of Lane and Vehicle Model. (a) Lane image before background subtraction. (b) Lane image after background subtraction showing vehicle statistical model.

2.2 The Per Pixel Detection Model

The background subtracted lane image $\tilde{\phi}_{ij}(t)$ is modeled as a three component Gaussian mixture. The central peak in the mixture is centered at zero and represents the background and the two other peaks (typically much smaller than the central one) are due to vehicles. The component of positive mean corresponds to bright points in a vehicle and the component of negative mean corresponds to shadows.

These considerations lead to the following simple classification of individual pixels in the (background subtracted) lane image:

$$\begin{array}{lll} H_0: & X \sim \mathcal{N}(0,\sigma^2) & \text{[background]} \\ H_b: & X \sim \mathcal{N}(\mu_b,\sigma^2) & \text{[bright]} \\ H_d: & X \sim \mathcal{N}(\mu_d,\sigma^2) & \text{[dark]} \end{array}$$

where X denotes a generic pixel (i.e., one of the $\tilde{\phi}_{ij}(t)$). If the relative proportions of bright and dark pixels were known it would be possible to formulate a two class Neyman–Pearson test for H_0 versus a combination pixel distribution representing a vehicle

$$\frac{1}{\sqrt{2\pi}\sigma}\left[\pi_d \exp\left\{-\frac{(x-\mu_d)^2}{2\sigma^2}\right\} + \pi_b \exp\left\{-\frac{(x-\mu_b)^2}{2\sigma^2}\right\}\right].$$

The difficulty in this approach is that the mixing terms π_b and π_d are unknown and will typically vary from frame to frame as vehicles leave and enter the field of view or change in apparent size.

Instead, the approach taken here is to separately test H_0 versus H_b and H_0 versus H_d via Neyman–Pearson detectors and to combined the results so as to avoid using knowledge of the mixing proportions. Separately, these problems amount to Gaussian location testing for which it is straightforward to show that the α_b level Neyman Pearson test for H_0 versus H_b selects hypothesis H_b if the pixel $X \geq \tau = \sigma\Phi^{-1}(1-\alpha_b)$. The resulting detection probability[1] is $1-\Phi(\Phi^{-1}(1-\alpha_b)-\mu_b/\sigma)$. Similarly, for the H_0 versus

[1] $\Phi(x)$ denotes the cumulative probability distribution function of a $\mathcal{N}(0,1)$ random variable.

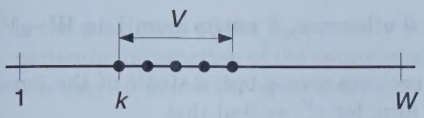


Figure 2: Idealized Line Image. Vehicle position is indicated by dark circles.

H_d problem the Neyman–Pearson test is of the form $X \leq \eta = \sigma\Phi^{-1}(\alpha_d)$ and the detection probability is $\Phi(\Phi^{-1}(\alpha_d) - \mu_d/\sigma)$.

A pixel by pixel classification could be done by a simple two sided thresholding of the value of each pixel in the image $\tilde{\phi}_{ij}(t)$. If the value fell between η and τ the pixel would be declared background and otherwise vehicle. Such a method would obviously require some spatial filtering because the performance of the pixel by pixel classification would not typically be acceptable. An alternate approach would be to incorporate prior knowledge of the shape of vehicles into the model. That is, that vehicle pixels come grouped together as shown in Figure 1. However, even retaining the assumption of independent pixels, such would entail a composite testing problem with three degrees of freedom (location and size of vehicle) and the resulting search would be prohibitive. In the next section, we make a further refinement of the vehicle model which concentrates only on lines of the lane image to simplify considerably the computational effort involved.

2.3 The Per Line Detection Model

To simplify the complexity of the proposed vehicle detector we have modified the model slightly to concentrate on individual lines in the lane image. Lines will be placed into one of three classes:

$\bar{H}_0$: line intersects background.
$\bar{H}_b$: line intersects bright part of vehicle.
$\bar{H}_d$: line intersects dark part of vehicle.

Classification will be done by first testing $\bar{H}_0$ versus $\bar{H}_b$. If the bright hypothesis is selected the line is declared to have intersected a vehicle. Otherwise, we test $\bar{H}_0$ versus $\bar{H}_d$. If the dark hypothesis is selected the line is then declared to have intersected a vehicle. If both the bright and the dark hypotheses fail, the line is declared to be a part of the background. For the description to follow we will concentrate solely on the $\bar{H}_0$ versus $\bar{H}_b$ test. Under all of the above hypotheses, the pixels in the selected line are modeled as independent. Fix a frame number t and a particular line i and suppose that the lane is W pixels wide at line i. The random variables observed in this line of the image are (see Figure 2) $\{\tilde{\phi}_{ij}(t) : 1 \leq j \leq W\}$ but they will be denoted $\{X_j : 1 \leq j \leq W\}$ for simplicity. Then the density for hypothesis $\bar{H}_0$ is

$$\bar{p}_0(\underline{x}) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^W \exp\left\{-\frac{1}{2\sigma^2}\sum_{k=1}^{W} x_k^2\right\}.$$

Hypothesis $\bar{H}_b$ is composite in that both the vehicle width V and position k are unknown. Thus it consists of the family of densities

$$\bar{p}_k^V(\underline{x}) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^W \exp\left\{-\frac{1}{2\sigma^2}\sum_{j=1}^{W}(x_j - \mu_b w_k^V(j))^2\right\}$$

where $w_k^V(j) = 1$ if $k \leq j \leq k + V - 1$ and $= 0$ otherwise, k ranges from 1 to $W - V$, and V ranges over allowable vehicle widths.

A generalized likelihood ratio test for this problem uses a test statistic of the form: $\bar{L}_b(\underline{X}) = \max_{V,k} \bar{p}_i^V(\underline{X})/\bar{p}_0(\underline{X})$. Expanding the form for $\bar{p}_k^V$ we find that

$$\bar{p}_k^V(\underline{x}) = \bar{p}_0(\underline{x}) \exp\left\{\frac{\mu_b}{\sigma^2}\left[\sum_{j=1}^{W} w_k^V(j)x_j - \frac{\mu_b V}{2}\right]\right\}$$

and, therefore, the test statistic may be written

$$\bar{L}_b(\underline{X}) = \max_{V,k} \exp\left\{\frac{\mu_b}{\sigma^2}\left[\sum_{j=1}^{W} w_k^V(j)X_j - \frac{\mu_b V}{2}\right]\right\}.$$

The Neyman–Pearson optimal test selects hypothesis $\bar{H}_b$ when $\bar{L}_b(\underline{X})$ is greater than a specified threshold and this is easily seen to be equivalent to comparing

$$\max_{V,k}\left\{2\sum_{j=1}^{W} w_k^V(j)X_j - \mu_b V\right\} \tag{3}$$

to a related threshold. This test is unacceptable for two reasons. First, it requires that μ_b be known in order to set the threshold mentioned above. However, the means of the two vehicle distributions (bright and dark) may be difficult to estimate given the relative paucity of vehicle pixels in comparison to background pixels. Second, the distribution of the statistic in Equation (3) will be difficult to calculate or approximate directly.

Hence, we further approximate by using a fixed and typical value for V which we assume is an integer fraction of the lane width, i.e., we take $W/V = N$. Further, we restrict the possible vehicle locations in the lane line to be of the form $k = (n-1)V + 1$ where $1 \leq n \leq N$. If we define

$$Y_n = \sum_{j=(n-1)V+1}^{nV} X_j$$

then we may use the follow approximate statistic in place of (3)

$$Z_b = \max_{1 \leq n \leq N} Y_n. \tag{4}$$

This approximation removes the dependence upon the bright mean μ_b and allows the computation of the distribution of the test statistic since the variables Y_n are independent under either hypothesis.

Under $\bar{H}_0$ the Y_n are iid and distributed individually as $\mathcal{N}(0, V\sigma^2)$. It is easy to show then that the distribution of Z_b under $\bar{H}_0$ is $\Phi^N(z_b/\sqrt{V}\sigma)$. Hence, for a specified false alarm rate $\bar{\alpha}_b$ the Neyman–Pearson test for $\bar{H}_0$ versus $\bar{H}_b$ is

$$Z_b = \max_{1 \leq l \leq N} Y_l \quad \begin{matrix} \geq \\ \\ < \end{matrix} \quad \tau = \sqrt{V}\sigma\Phi^{-1}((1-\bar{\alpha}_b)^{1/N}) \quad \begin{matrix} \text{decide } \bar{H}_b. \\ \\ \text{decide } \bar{H}_0. \end{matrix} \tag{5}$$

The approximation made in replacing statistic (3) with (4) amounts to replacing the original composite hypothesis $\bar{H}_b$ with a family of distributions for the Y_n (one for each $1 \leq m \leq N$), specifically, $Y_m \sim \mathcal{N}(V\mu_b, V\sigma^2)$, $Y_n \sim \mathcal{N}(0, V\sigma^2)$ for $n \neq m$. Since Z_b

is the maximum of the Y_n we easily see that its distribution will not depend upon the particular incarnation of the approximate $\bar{H}_b$ hypothesis above. Arguing along these lines it is easy to derive the approximate performance of the detector (5)

$$\begin{aligned}\bar{P}_b\{Z_b > \tau\} &= 1 - \Phi\left(\frac{\tau - V\mu_b}{\sqrt{V}\sigma}\right)\Phi^{N-1}\left(\frac{\tau}{\sqrt{V}\sigma}\right) \\ &= 1 - \Phi\left(\Phi^{-1}((1-\bar{\alpha}_b)^{1/N}) - \sqrt{V}\frac{\mu_b}{\sigma}\right)(1-\bar{\alpha}_b)^{(N-1)/N}.\end{aligned}$$

Since the minimum of the Y_l's is the maximum of the $-Y_l$'s, the test of $\bar{H}_0$ versus $\bar{H}_d$ is similar.

2.3.1 Combining the Bright and Dark Line Hypotheses

The combined line hypothesis test can be summarized as follows. If the minimum Z_d is less than η or the maximum Z_b is greater than τ declare that the line includes vehicle pixels. Otherwise, declare the line to be part of the background. The false alarm rate for this combined test would be $\bar{\alpha} = \bar{P}_0(\{Z_d < \eta\} \cup \{Z_b > \tau\}) = \bar{\alpha}_d + \bar{\alpha}_b$ since the events above are disjoint (note: $\eta < 0 < \tau$). In order to attempt the optimization of $\bar{\alpha}_d$ and $\bar{\alpha}_b$ subject to the overall false alarm constraint we would need to know mixing proportions for the bright and dark hypotheses (say $\bar{\pi}_b$ and $\bar{\pi}_d$). The per line vehicle hypothesis could be viewed as a sum $\bar{P}_v = \bar{\pi}_b\bar{P}_b + \bar{\pi}_d\bar{P}_d$ though it is important to rememember that all of the above are actually composite distributions. Then we could approximate the performance of the combined test by

$$\begin{aligned}\bar{P}_v(\{Z_d < \eta\} \cup \{Z_b > \tau\}) &= \bar{P}_v\{Z_d < \eta\} + \bar{P}_v\{Z_b > \tau\} \\ &\approx \bar{\pi}_d\bar{P}_d\{Z_d < \eta\} + \bar{\pi}_b\bar{P}_b\{Z_b > \tau\}.\end{aligned}$$

2.4 Combination of the Per Line Tests

The final step of the detection process is to combine the line-by-line results into vehicle detections. Recall that the result of a line test is 0 (background) or 1 (vehicle, either bright or dark). As shown in Figure 3, the line-by-line results have both false positive and missed detection errors. However, after processing by a 1-dimensional 21-point median filter, the errors are removed. The length of the filter was determined as half of the length of a small vehicle (i.e., motorcycle) in the rear of the image.

3 Dynamic Model and Observation Geometry

3.1 Vehicle Dynamics

In most multiple target tracking problems where computational complexity is an issue, target dynamics are handled with relatively simple (two or three state) models. The motivation for this is twofold. First, often the measurement to track association problem is quite difficult and requires considerable computational resource. Second, simple dynamic models frequently suffice particularly when good coordinate systems have been chosen and/or it is inadvisable to overmodel imprecisely known dynamics. These issues are also important in highway monitoring especially given the tightness of computational resources in an economically viable system.

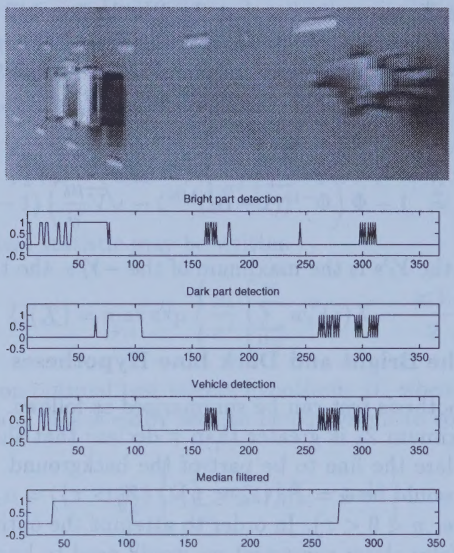


Figure 3: The Detector: Intermediate and Final Results.

Our approach is to model the dynamics of individual vehicles separately, that is, the model will assume no coupling between vehicles either in a single lane or between adjacent lanes. Therefore, vehicles are assumed to remain in their lane and lane changing activity is outside of the model and is handled by a supervisory track–observation association function. Within a lane we assume, for dynamic modeling purposes, that the vehicle follows a know path so down-road (longitudinal) dynamics along the path but not cross-road (lateral) dynamics perpendicular to the path are modeled. We assume the standard constant velocity model which, when sampled, yields the discrete time model

$$\begin{bmatrix} x_{k+1} \\ \dot{x}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ \dot{x}_k \end{bmatrix} + w_k$$

where T is the observation sampling interval and the process noise w_k is assumed to be zero mean, white, with covariance matrix

$$Q = \mathrm{E}\{w_k w_k^T\} = \begin{bmatrix} T^3/3 & T^2/2 \\ T^2/2 & T \end{bmatrix} q, \quad q > 0.$$

More detailed dynamic models based on human factors are available [2] which include interactions between vehicles in one lane and the work described here could, at the cost of additional computational complexity, be extended to use such models.

3.2 Road–Camera Geometry

The fundamental input to the MTT is a detection: "vehicle at pixel (i, j)". The image coordinates $\vec{x}^I \in \mathcal{R}^2$ of a location $\vec{x} \in \mathcal{R}^3$ within the field of view of the camera are $\vec{x}^I = h(\vec{x})$ where h contains a linear rotation and translation and a nonlinear projection [3]. Given that the vehicle is constrained to lie on the road, h can be inverted to compute road

location $\vec{x}$ from image location $\vec{x}^I$. In the MTT we use road coordinates in the tracking filters and an observation that is h^{-1} applied to the return provided by the detection algorithm. This results in an observation equation with a state-dependent observation noise covariance: $z_k = Hx_k + \epsilon_k$ where $H = [1\ 0]$, $\mathrm{Cov}[\epsilon_k] = R_k(x_k)$, and $z_k = h^{-1}(\vec{x}_k^I)$.

4 Multiple Target Tracking

The fundamental issue in MTT is data association—which return is due to which target, including the possibility that some returns do not correspond to any target (clutter) and that the return from some targets at some times may be missed. Our approach has 2 steps. First, we select out a set of candidate returns based on a gate determined from the standard deviation of the predicted location provided by the dynamic system model. Second, since most of the clutter in this problem is due to end-of-vehicle effects, we use a nearest neighbor filter which selects the return closest to the predicted location. This is equivalent to a Generalized Likelihood Ratio test in which the parameter is the identity of the return. There are more sophisticated methods, for instance, methods which ensure that no return is associated with more than one target or methods which wait until $n + n_0$ before making associations at step n (delayed pruning), but we have found that this simple approach is sufficient for the traffic situations we have considered.

We deal with lane changes in a post-processing stage. After performing per-lane data association for each lane, those returns which have not been associated with any track are determined. For this set of returns, a second round of gating and association are performed where the predicted locations are generated by tracks which, at the previous time-step, were in adjacent lanes. If the association is successful, then the vehicle is declared to have changed lanes.

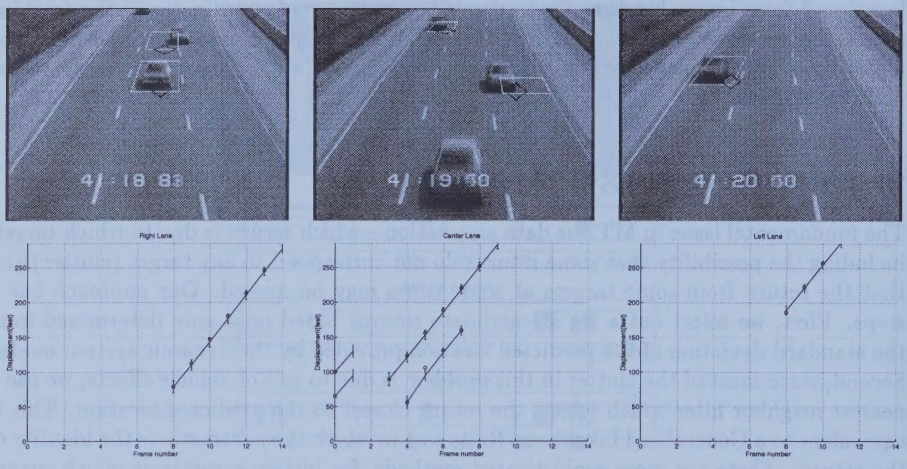
In addition to performing track continuation, as described above, an MTT must also perform track initialization, confirmation, and deletion. We use simple algorithms: A track is initialized whenever an return cannot be associated with any existing track. A track is confirmed if an observation is associated with it at the sample time following its initialization. A track is deleted if for 2 consecutive sample times there has been no return associated with the track.

5 Results

Figure 4 shows 3 frames from a video sequence. Rectangular (diamond) boxs mark the detected (filtered) position. The height of the diamond corresponds to the standard deviation of the position estimate from the Kalman filter. Below the 3 images are 3 space-time plots, one plot for each lane. Tracks and detections are represented by solid lines and circles respectively. Error bars correspond to the standard deviation of the tracks' positions. Notice the one vehicle which changes lanes so that it's track terminates in the center-lane space-time plot and continues in the left-lane plot.

References

[1] P. Bouthemy and P. Lalande. Recovery of moving object masks in an image sequence using local spatiotemporal contextual information. *Opt. Eng.*, 32(6):1205–1211, June 1993.



(a) (b) (c)

Figure 4: Tracking Results: Three Selected Images and Per-Lane Space-Time Trajectories for the (a) Right, (b) Center, and (c) Left Lanes.

[2] R. E. Chandler, R. Herman, and E. W. Montroll. Traffic dynamics: Studies in car following. *Operations Research*, pages 165–184, March 1958.

[3] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* Wiley-Interscience, New York, 1973.

[4] Federal Highway Administration Contract DTFH61–91–C–0076, U.S. Department of Transportation, Washington, D.C. *Detection Technology for IVHS: Task F Report – Vehicle Detector Field Test Specifications and Field Test Plan*, 1993.

[5] A. L. Gilbert, M. K. Giles, G. M. Flachs, R. B. Rogers, and Y. H. U. A real-time video tracking system. *IEEE Trans. Pattern Anal. Mach. Intell.*, PAMI-2(1):47–56, January 1980.

[6] D. P. Looze, P. K. Houpt, J. N. R. Sandall, and M. Athens. On decentralized estimation and control with application to freeway ramp metering. *IEEE Trans. Automatic Contr.*, AC-23(2):268–275, April 1978.

[7] P. G. Michalopoulos. Vehicle detection video through image processing: The autoscope system. *IEEE Trans. Veh. Technol.*, VT-40:21–29, February 1991.

[8] D. D. Sworder and R. G. Hutchins. Image-enhanced tracking. *IEEE Trans. Aero. Elec. Sys.*, AES-25(5):701–709, September 1989.

[9] U. S. Dept. of Transportation, Washington, D. C. *Department of Transportation's IVHS Strategic Plan - Report to Congress*, December 1992.

Multivariable Control of Plasma Enhanced Chemical Vapor Deposition Using Real-Time Mass Spectroscopy

T.J. Knight, X. Cheng, D.W. Greve, B.H. Krogh, M.A. Gibson
Dept. of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213-3890
e-mail: krogh@cmu.edu

Abstract

This paper reports on the design and implementation of an experimental plasma enhanced chemical vapor deposition (PECVD) system, and initial experiments for system identification and multivariable control. The primary objective of this research is to develop methods for using new *in situ* sensors to directly control values of process variables that are more directly related to the quality of critical processes for integrated circuit (IC) fabrication than the manipulated variables that are typically controlled in current practice. The new sensing used in the experimental PECVD system is a real-time mass spectrometer that provides measurements of critical species concentrations in the plasma impinging on the substrate surface. A three-input, three-output model of the linearized process dynamics, derived from system identification experiments, was used to design a linear-quadratic optimal controller and a Kalman filter to achieve setpoint control of two species concentrations and the plasma bias voltage. Simulation results are presented for multivariable control of these three process variables. The paper concludes with a discussion of the next steps in this research project.

1. Introduction

Process control is of central importance in the semiconductor industry. The manufacture of a semiconductor device requires numerous unit processes performed in sequence, where each unit process presents a formidable process control problem. As integrated circuits have become more complex, the performance requirements for each unit processes have become more demanding. Consequently, major efforts have been made to increase the reproducibility of semiconductor processes. These efforts have included continual improvements in process equipment design and advances in raw material production to reduce faults caused by impurities. In addition, statistical process control (SPC) techniques have become widely used to monitor processes and initiate corrective action when an out-of-tolerance condition is detected (Mozumber and Strojwas, 1990).

Figure 1 illustrates the way in which unit processes are controlled in current practice using SPC techniques. The *process recipe* defines the process operating point in terms of setpoint values for the variables that are directly controlled by single-loop PID (proportional-integral-derivative) controllers. These *manipulated variables* (gas flows, temperatures, pressures, power, etc.) are maintained at values prescribed by the process recipe which is determined by extensive off-line experiments. The results of the unit step are characterized by *performance variables* which characterize the device-related physical properties that will influence the ultimate performance of the IC being fabricated. Performance variables are often measured after a unit process is completed, and this information is used in the SPC technique to tune the process recipe to improve the unit process performance for future batches.

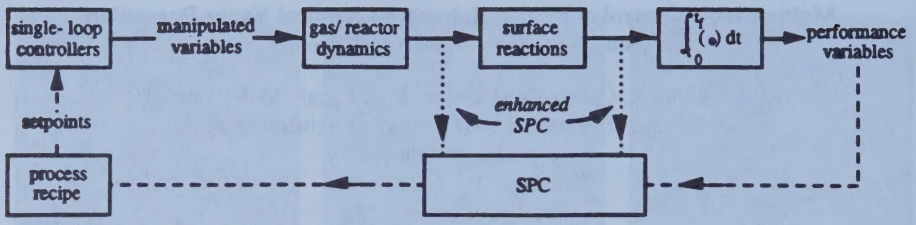


Figure 1. Conventional control of a semiconductor process.

Despite the success of SPC in dealing with long-term process drift, conventional techniques cannot compensate for progressive drift that occurs during the operation of the process. Recently, measurements from *in situ* sensors have been incorporated into SPC methods to achieve more direct control of citric process variables (identified as *enhanced* SPC in Fig. 1), but this additional information is still only used to make adjustments after the process step is completed for the current batch of wafers.

The objective of the research project described in this paper is to use measurements from *in situ* sensors to regulate the unit process in real time. In this new process control architecture, illustrated in Fig. 2, the process recipe is new given in terms of *process variables* that are more directly related than the manipulated variables to the ultimate values of the performance variables. Control of the process is still effected through the single-loop controllers for the manipulated variables. The setpoints for the manipulated variables are varied on line, however, by a new multivariable controller designed to maintain the process variables at the values specified in the process recipe. As indicated in Fig. 2, SPC techniques could be applied in this new scheme to achieve batch-to-batch control of long-term drift by tuning the new process recipe. The focus of the work reported in this paper, however, is on the design of the

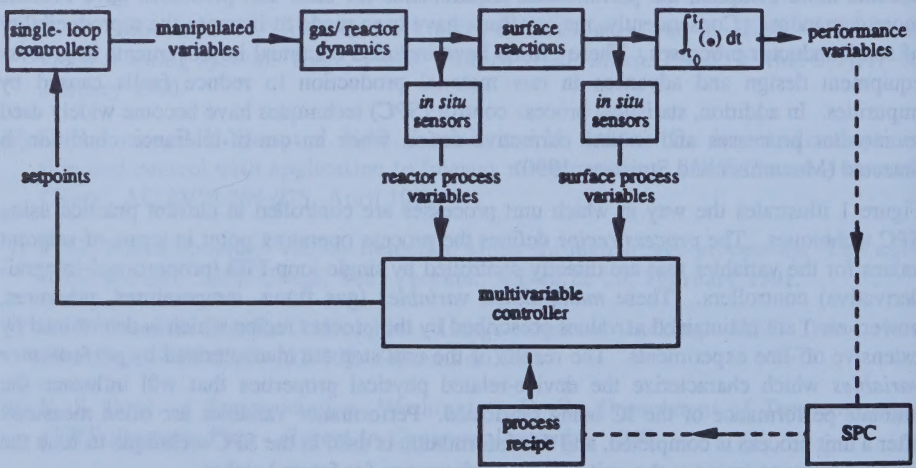


Figure 2. Implementation of process variable control using *in situ* sensors.

multivariable controller for real-time control of a single plasma deposition step. Similar work has been reported for plasma etching (Butler et al., 1991; Elta at al., 1993; McLaughlin et al., 1991).

To achieve control of process variables using *in situ* sensors, there are a number of issues that need to be addressed. First, the new sensors need to be incorporated into the process equipment and experiments need to be performed to determine the quality and sampling frequency of the new measurements. The process then needs to be characterized in terms of the new process variable measurements. This includes both steady state operating points to create the process recipes, and models of the process dynamics characterizing the input-output relationships between the manipulated variables and the process variables. As indicated in Fig. 2, these input-output relationships are determined by the gas and reactor dynamics as well as the surface reactions. Since semiconductor processes are not easily modeled from first principles, and in some cases are not well understood, process models must be developed and verified using experimental data and system identification techniques. The process models are used to design the multivariable controller. Since there are many constraints and physical considerations that need to be taken into account in the controller design, simulation studies are required to verify and tune the controller design before it is implemented on line. The final step is to implement the real-time controller and obtain experimental verification of the performance.

In the remainder of this paper we present the initial work on carrying out the general program described above for a PECVD system with new *in situ* sensing of species concentrations using a real-time quadrupole mass spectrometer. The particular instrument chosen (MKS PPT-600A) is one of a new class of sensors which are compact, self-contained, and offer direct computer interfacing. PECVD systems are widely used in semiconductor manufacturing to deposit insulators for passivation and inter-layer dielectrics. The complexity of the process precludes the development of accurate process models based on physics and chemistry alone. Moreover, quadrupole mass spectrometers have only rarely been applied as sensors for closed-loop control (van de Leur et al., 1988; Heyn, 1993), and to our knowledge they have never been applied in a multivariable control system. Consequently, experimentation and interpretation of experimental data plays a key role in successfully controlling process variables in the manner described above. Results reported in the literature suggest that certain species concentrations can be critical process variables in plasma deposition (Smith et al., 1990). The preliminary results presented in this paper demonstrate the feasibility of accomplishing real-time regulation of key process variables.

The following section describes the custom experimental PECVD system that has been implemented in our laboratory. In section 3 we describe several experiments that have been performed to establish the basic operating characteristics of the sensor and identify an appropriate transfer function matrix for our initial multivariable control system design. The controller design and simulation results are presented in section 4. The concluding section discusses the issues to be addressed in the next stages of this research project.

2. The PECVD System

Plasma-deposited amorphous silicon nitride (Si_3N_4) and silicon dioxide (SiO_2) are important materials for IC fabrication. Plasma excitation permits deposition at low substrate temperatures ($\approx 300^oC$). Performance variables for insulator deposition include breakdown voltage, thickness, dielectric constant, stress and uniformity. In a conventional PECVD reactor, the manipulated variables are the mass flows of the incoming gases, the substrate

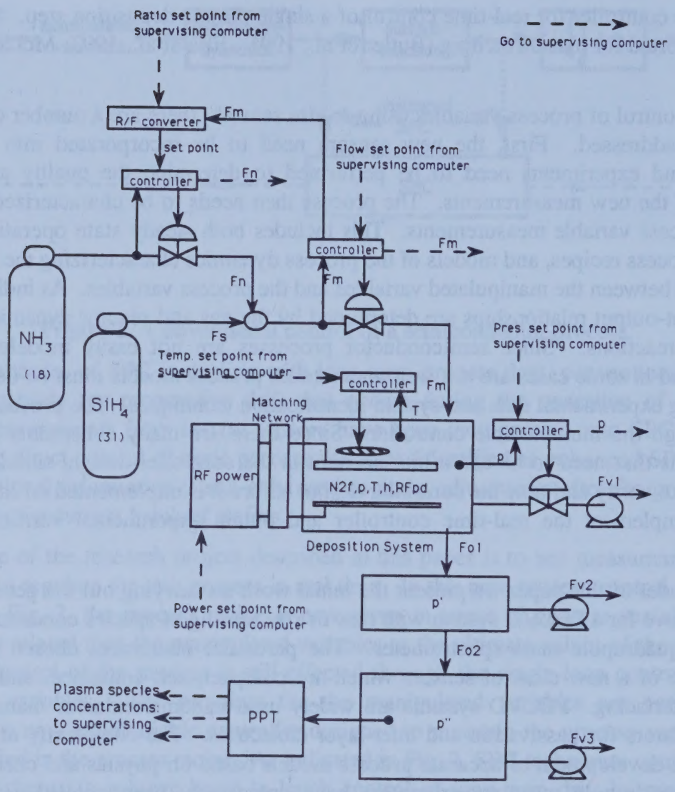


Figure 3. Experimental plasma enhanced chemical vapor deposition system.

temperature, the RF power applied to the plasma, and the chamber pressure. Each of these manipulated variables is sensed and controlled to specified setpoints with independent single-loop controllers. Supervisory control to implement the process recipe may be accomplished manually or automatically from a supervisory control computer. There is no real-time adjustment of the setpoints for the manipulated variables during a process run, however, because no other process variables are measured in a conventional system.

We have designed and built a new PECVD system in our cleanroom that incorporates a differentially-pumped quadrupole mass spectrometer to sample species directly incident on the wafer platform during a process run. By sampling on the wafer platform, the complications of gas-phase reactions associated with down-stream sampling are avoided. In addition, there is the potential for direct measurement of the radical species responsible for deposition. As illustrated in Fig. 3, this *partial pressure transducer (PPT)* provides real-time measurements of the species concentrations which can be used by the supervisory control computer to implement process recipes defined in terms of these concentrations. Regulation of selected species concentrations and the plasma DC bias, which is available as a measurement from the RF power controller in our system, can be accomplished by adjusting the setpoints for the manipulated variables as illustrated in Fig. 2. The substructures are placed on a 17cm

diameter heated platform with recesses for two 75mm wafers. Gases are supplied through a showerhead in the upper electrode with a spacing of 2.7cm.

Our supervisory control computer is a Gateway 486/66 running LabWindows/CVI. Temperature control is accomplished through an independent single variable controller, whose setpoint is set from the control computer. The total pressure of the system and the flow rates of silane and ammonia are controlled by single variable controllers packaged together in a multi-gas controller (MKS 647A). RF power is controlled by an RF generator/controller (Advanced Energy ATX/RFX-600), which also measures the plasma bias voltage. The temperature controller and the PPT are connected to the computer via RS-232 at 2400 Baud and 19,200 Baud, respectively. This limits the maximum sampling speed of the temperature controller to 10 Hz. The PPT has internal limitations due to the physical measurement process, and thus is limited to approx. 1 Hz operation. The multi-gas controller uses IEEE-488 communication, but is limited by measurement considerations to 20 Hz. The RF power controller uses analog inputs and outputs from the computer.

3. System Identification Experiments

A number of experiments have been performed to determine the characteristics of the PPT and to calibrate and adjust the sensors, actuators, and controllers in the PECVD system. Based on the timing characteristics of the instruments and data communication links in the system, it was determined that the supervisory control computer could perform full data acquisition and control at a frequency of 1 Hz. We have also performed some initial deposition experiments. In this section we describe the results of experiments performed to obtain a multi-input multi-output transfer function model of the process dynamics for some selected variables at a specific operating point to implement an initial multivariable controller. The objective in this particular effort was to complete the full controller implementation and verify its operation as a preliminary test of the overall control architecture illustrated in Fig. 3. The particular operating point and selected species were chosen more for expediency in than practical significance in this initial set of experiments and implementation.

Step response experiments were performed to determine the input-output transfer functions between of pair of selected manipulated variables and process variables. The results of one of these experiments shown in Fig. 4 illustrates the type of drift that can occur when the manipulated variables are held at constant setpoint values as is done typically in the conventional process control systems (Fig. 1). Our initial data acquisition shows the day to day variation in output species is about +/-30% of average steady state value. Possible causes of this drift are deposition of material on the internal surfaces and varying quantities of water vapor. In conventional systems, this drift would not be measured in real time and its effects on the deposition would only be determined after the process run is over, making it too late to compensate for the problems that might arise in the current batch of wafers.

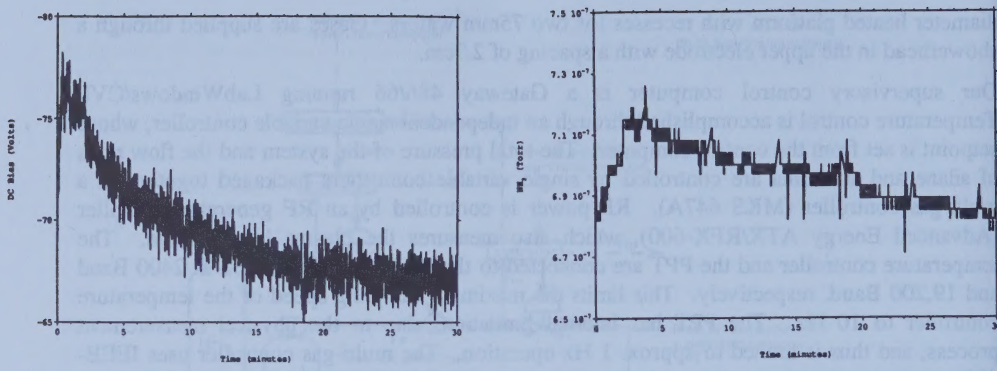


Figure 4. Process variables drift during one experimental run (operating point: NH_3 flow 40 sccm, SiH_4 flow 2 sccm, RF Power 20W, Pressure .15 Toor).

Fig.5 shows the step response leading to the identified transfer function matrix. . The input output data obtained from the step response experiments was analyzed using the MATLAB System Identification Toolbox to obtain the transfer function matrix:

$$G(s) = \begin{array}{c} \text{Input:} \\ \\ \\

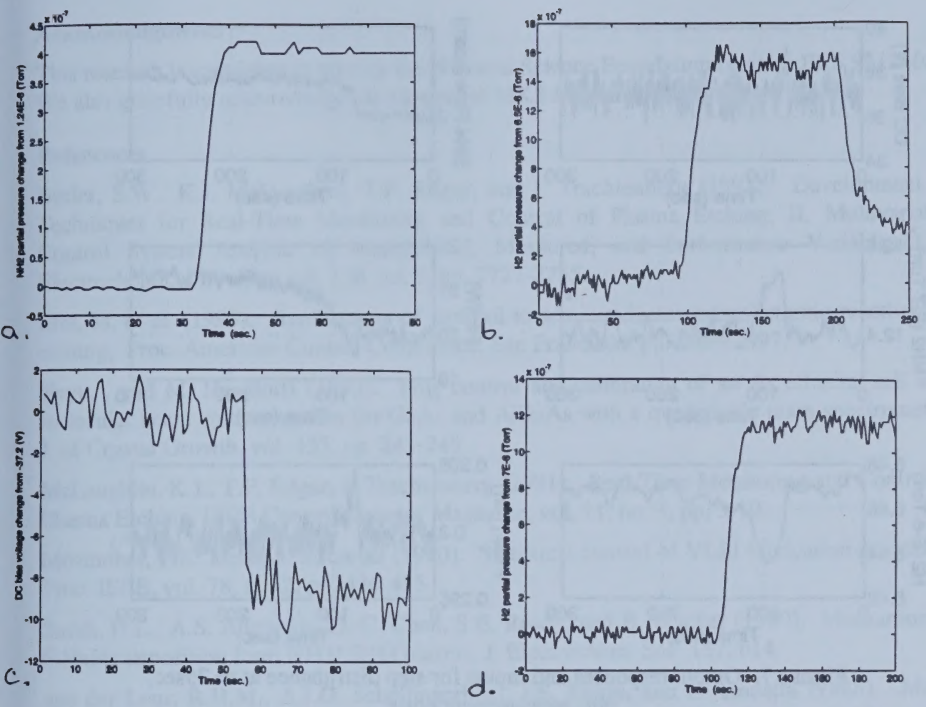


Figure 5. Step responses of process variables: a. 30% pressure step at t=32 sec; b. 50% SiH_4 steps at t=98 and 200 sec ; c. 25% RF step at t=55 sec; d. 25% NH_3 step at t=102sec.

where $\boldsymbol{x}$ is the state vector for the linearized system dynamics (1) plus the three integrators, $\boldsymbol{u}$ is the vector of three manipulated variable setpoint perturbations from the nominal operating point (the control input for the linearized dynamics), and **Q**,**R** are the symmetric weighting matrices used to tune the controller to achieve an acceptable response.

In our simulation study, the process noise and disturbance are chosen the same order of magnitude as in experimental data from the real process. Also modeling error between true

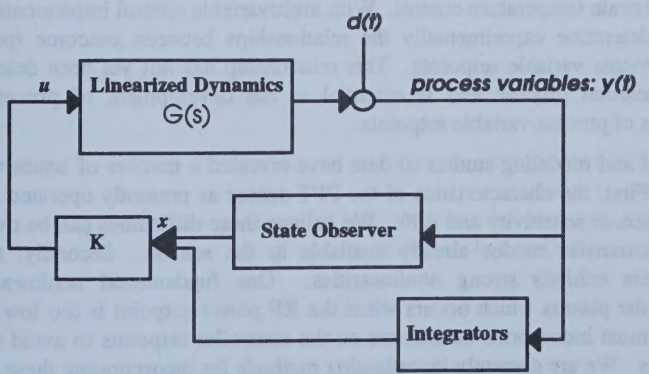


Figure 6. Linear multivariable controller design.

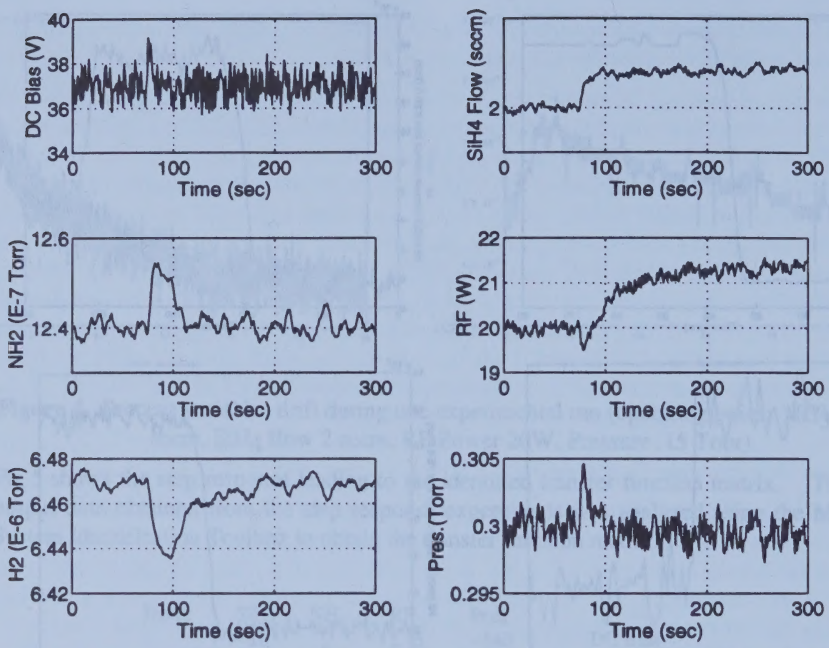


Figure 7. Output responses and inputs for step distrubance at t= 75sec, plus measurement noise.

plant and nominal model has been taken into account. The simulation results in Fig.7 show that output process variables are regulated well around setpoint values in the presence of noise and a constant disturbance imposed at 75 second.

5. Project Status and Future Research

At present, we are ready to begin testing of the multivariable controller designed on the basis of our system identification experiments. Our immediate objectives are to implement control of particular process variables (probably including the NH2 and H2 species discussed above) together with substrate temperature control. With multivariable control implemented, the next objective is to determine experimentally the relationships between outcome (performance variables) and process variable setpoints. This relationship has not yet been determined for this plasma deposition process and is essential in the development of process recipieS specified in terms of process variable setpoints.

Our experimental and modeling studies to date have revealed a number of issues which need to be resolved. First, the characteristics of the PPT sensor as presently operated has caused difficulties with sensor sensitivity and drift. We believe these difficulties can be overcome by using other data-transfer modes already available in the sensor. Secondly, the plasma deposition process exhibits strong nonlinearities. One fundamental nonlinearity is the extinguishing of the plasma which occurs when the RF power setpoint is too low. A robust controller, then, must incorporate constraints on the controller setpoints to avoid undesirable operating regimes. We are currently investigating methods for incorporating these constraints into the multivariable controller design.

Acknowledgments

This research is supported in part by the National Science Foundation (grant # ECS 9312406). We also gratefully acknowledge the support of MKS Instruments.

References

Butler, S.W., K.J. McLaughlin, T.F. Edgar, and I. Trachtenberg (1991). Development of Techniques for Real-Time Monitoring and Control of Plasma Etching, II. Multivariable Control System Analysis of Manipulated, Measured, and Performance Variables,Ó J. Electrochemical Society, vol. 138, no. 9, pp. 2727-2735.

Elta, M. et al. (1993). Applications of control to semiconductor manufacturing reactive ion etching, Proc. American Control Conference, San Francisco, pp. 2990-2997.

Heyn, , and M. Harsdorff (1993). Flux control and calibration of an As effusion cell in a molecular beam epitaxy system for GaAs and AlGaAs with a quadrupole mass spectrometer, J. of Crystal Growth, vol. 133, pp. 241-245.

McLaughlin, K.J., T.F. Edgar, I. Trachtenberg (1991). Real-Time Monitoring and Control in Plasma Etching, IEEE Control Systems Magazine, vol. 11, no. 3, pp. 3-10.

Mozumber, P.K. and A.J. Strowjas (1990). Statistical control of VLSI fabrication processes, Proc. IEEE, vol. 78, no. 2, pp. 436-455.

Smith, D.L., A.S. Alimonda, D.-C. Chen, S.E. Ready, and B. Wacker (1990). Mechanism of SiHxHy deposition form NH3/ SiH4 plasma, J. Electrochem. Soc. 137, 614.

van der Leur, R.H.M., A.J.G. Schellingerhout, J.E. Mooji, and F. Tuinstra (1988). Mass-spectrometer-controlled fabrication of Si/Ge superlattices, Appl. Phys. Lett., vol. 52, no. 12, pp. 1005-1007.

Control of Photoresist Properties: a Kalman Filter Based Approach *

Evan Palmer†, Wei Ren‡, Costas Spanos‡, and Kameshwar Poolla†

Abstract

The photolithography step in the semiconductor manufacturing process becomes increasingly critical as linewidths decrease for the next generation of integrated electronics. It therefore becomes necessary to reduce variations in photoresist parameters such as resist film thickness and photoactive compound concentration during manufacture. In this paper, we present a simple feedback scheme for accurately regulating these parameters. Our approach involves obtaining a simple, static process model the coefficients of which are recursively adjusted based on previous wafer measurements. We use this adaptive model to determine appropriate input setpoints for the next wafer. The effectiveness of this scheme in reducing process drift is exhibited by experimental data. Our results supports the widespread contention that modern feedback control offers the promise of improving the semiconductor manufacturing processes, often with relative ease and minimal capital cost.

1 Introduction

As semiconductor devices become increasingly complex, the ability to pack individual devices closer together on a single chip becomes very important. One of the key issues involved is the linewidth or critical dimension of each of these devices. Currently, the industry is successfully producing devices with CD's of 0.5μm. However, industry analysts are predicting that by the end of the century, CD's will have shrunk to less than half this size [2].

In order to successfully produce devices with significantly reduced linewidths, processing techniques must improve. This means reducing the process variation in all phases of production, especially those affecting lithography. We have focused our attention on two such sources of linewidth variation. Both resist thickness and photoactive compound concentration can drift as a batch of wafers is processed. It is our assertion that modern feedback control techniques are an inexpensive, effective means of reducing process variation and increasing throughput.

The experimental component of this work was conducted in the Microfabrication Laboratory at the University of California, Berkeley.

*Supported in part by the Semiconductor Research Corporation under Contracts 93-MP-700 and 94-YP-700, by the National Science Foundation under Grant ECS 89-57461, and by gifts from Rockwell International.

† Department of Mechanical Engineering, University of California, Berkeley, CA 94720.

‡ Department of Electrical Engineering, University of California, Berkeley, CA 94720.

The remainder of this paper is organized as follows. In Section 2, we describe the spin coating process in some detail. This includes a discussion of the control objectives, sensors and actuators, and process disturbances. Following this, in Section 3, we describe our control-oriented process model and the control strategy we employ. Section 4 offers details on the experiments we have conducted, and in Section 5 we summarize our results and draw conclusions.

2 The Spin Coating Process

In this section we describe the spin coating process in detail, outline the control objectives, discuss sensors and actuators, and the disturbances that affect the process.

2.1 Process Description

The spin coating process consists of several steps. The wafer is first pre-baked to dehydrate the wafer. The wafer is then coated with hexamethyldisilazane(HMDS). These initial steps increase adhesion of photoresist to the wafer surface. The application of photoresist to the wafer is done as follows. The wafer is spun at a relatively slow speed($\sim$500 rpm) while the resist is deposited from a nozzle whose position can be adjusted radially during the dispense. The wafer is then accelerated to a final spin speed, usually between 1500 and 6000 rpm, where it is held for somewhere between 30 and 40 seconds [15].
Once released from the spin chuck, the wafer moves to the soft-baking hot plate, where the bake process can last from 30 to 60 seconds. Bake temperatures are usually around 100 °C. After baking, the wafer is cooled on a cold plate, where the resist's thickness, uniformity, and photoactive compound concentration can be measured [15]. In our lab, the wafer track used was an SVG 8626 PC with an 8636 HPO bake plate. The resist that we used was OCG-820.

2.2 Control Objectives

In order to accommodate the finer linewidths that are called for in the next generation of integrated electronics it becomes critical to correctly set the depth of focus and the dose during exposure. These set points depend on resist thickness and photoactive compound concentration (PAC) respectively. It is therefore imperative to reduce variations in these parameters that arise in the spin-coating process. The focus of our paper is to employ feedback control methods to effectively and inexpensively achieve these objectives.

2.3 Sensors

We are (indirectly) able to measure both resist film thickness and PAC concentration. This fortunately happens to coincide with the choice of regulated variables made earlier. The equipment used for making our thickness and PAC concentration measurements was the Inspector, manufactured by SC Technology. This equipment consists of a photospectrometer, a xenon light source, and a 486 PC. The sensor tip is positioned above the cold plate and a measurement is taken whenever the PC receives a signal from the wafer track indicating that a wafer is in place. The sensor software takes the reflectance curves generated by the sensor and performs a curve fit at each end of the reflectance spectrum.

From these fits, the algorithm generates estimates for resist thickness and PAC concentration. Details of this fitting process can be found in [5]. The software requires three Cauchy values which describe the optical properties of the photoresist. For all of our experiments, we used $A = 1.91$, $B = 1.78 * 10^6$, $C = 1.218 * 10^{13}$ [8].

2.4 Actuators

In the context of our control objectives, the spin coating process is most significantly affected by spin speed and soft bake temperature. The spin speed directly affects the (pre-bake) resist thickness. The soft bake temperature affects both the final resist thickness (by evaporation of residual solvent) and the PAC concentration (by chemical breakdown of some of the photoactive compound). We have therefore chosen to actuate these variables to effect our control objectives.

The wafer track had a manufacturer supplied controller to regulate the spin speed of the motor on the spin chuck. This built in controller was only able to hold the spin speed to within $\pm 1\%$ of the desired speed. At 5000 rpm, this accuracy is ± 50 rpm. Strangely enough, the wafer track allows the user to specify the desired spin speed to within ± 10 rpm. We have, nevertheless, chosen to use the setup as is, without any hardware modifications to allow for finer spin speed regulation.

The soft-baking hot plate was heated by a resistive coil and had a manufacturer supplied controller that allowed the user to specify the temperature to within $\pm 0.1°C$. This accuracy was rarely utilized as the settling time necessary to achieve these tolerances was on the order of several minutes.

2.5 Disturbances

In addition to being affected by the actuated inputs, the spin coating process is affected (to a lesser degree) by such diverse factors as the humidity and temperature of the room, the viscosity of the resist, and wear and age of the wafer track. We collectively refer to these as process disturbances. Some of these we have no control over and others we choose not to actuate. For instance we could, at considerable expense, enclose the spin-coater in an environmentally regulated chamber.

3 System Identification and Control Strategy

We begin this section by discussing the structure of a multi-level, supervisory controller. Following this, we describe the control-oriented process models and the adaptive, Kalman Filter based control strategy we employ.

3.1 Control Strategy

There are three hierarchical levels of control that can and should be implemented on a spin coating system. At the highest level is a supervisory controller overseeing initial spin speed and bake temperature setpoints for batches of wafers all having the same desired thickness and photoactive compound concentration. Current work in statistical process control using response surface modelling is well suited for the design of these controllers [6, 13].

At an intermediate level is the run to run controller which determines the spin speed and bake temperature setpoints for each wafer that is processed. Initial command settings for this controller on a batch of wafers are determined by the supervisory controller. The run to run controller requires that inline measurements be made so that information about the current state of the process can be fed back to determine parameters for the next wafer. *The remainder of this paper focuses on the design of this run to run controller.*
The lowest level consists of real time control. In this context, the spin coating process could be regulated by shifting the spin speed in mid process based on real time thickness measurements. There are currently several companies developing products of this nature. Real time control becomes necessary only in the event that the run to run controller does not supply adequate performance. The (potential) performance enhancement supplied by real time control is at the expense of increased capital cost and equipment complexity.

3.2 Control-Oriented Process Modelling

The spin coating process is static or memoryless. Equivalently, the current values of resist thickness and photoactive compound concentration do not depend on input settings during previous runs.
To regulate resist thickness, we need a model relating the final spin speed ω (input) and the resist thickness t (output). To this end, a simple static input-output relationship of the following form may be found in several sources [3, 4]:

$$\log(t) = a\log(\omega) + b + w_k \tag{1}$$

Here w_k is measurement noise associated with resist thickness metrology for the kth wafer. This model is obtained by simplifying a fairly complex, non-Newtonian fluid dynamics based model of an idealized spin-coater.
In place of this simple model, we could have employed a very detailed model that incorporates disturbances such as resist viscosity and environmental effects. While this complex model would be excellent for simulation purposes, its utility for controller design would be limited. This is because controller complexity is usually comparable to model complexity, and an exceedingly complex controller would preclude reliable implementation.
A *time-invariant* static model of the form (1) would have poor predictive ability across a lot of wafers. This is because the simplified model (1) does not include process disturbances. We can nevertheless incorporate the effects of these disturbances by allowing the parameters a and b in the model (1) to drift slowly. This would improve the predictive ability of our model. We therefore choose to employ a model of the form

$$\log(t_k) = a_k\log(\omega_k) + b_k + w_k \tag{2}$$

where k is the wafer index. We can write the parameter drift as

$$\begin{bmatrix} a_{k+1} \\ b_{k+1} \end{bmatrix} = \begin{bmatrix} a_k \\ b_k \end{bmatrix} + v_k \tag{3}$$

where v_k is a two-dimensional parameter noise that accounts for process disturbances. In the absence of further information we model the noise sequences $\{v_k\}$ and $\{w_k\}$ as zero-mean, white, Gaussian random processes with covariance matrices Q and R respectively. We also make the reasonable assumption that the parameter noise and the measurement noise are statistically independent.

To simultaneously regulate resist thickness and PAC, we chose to employ a static model of the form

$$\begin{aligned} \log(t_k) &= a_k \log(\omega_k) + b_k \log(T_k) + c_k + w_k^{(1)} \\ \log(PAC_k) &= d_k \log(\omega_k) + e_k \log(T_k) + f_k + w_k^{(2)} \end{aligned} \tag{4}$$

Here the inputs are the bake temperature T_k and the final spin speed ω_k, the outputs are the resist thickness t_k and the photoactive compound concentration PAC_k, and k is the wafer index. As before we allow the model parameters $a_k, \cdots, f_k$ to drift slowly to incorporate disturbance effects as

$$\begin{bmatrix} a_{k+1} \\ \vdots \\ f_{k+1} \end{bmatrix} = \begin{bmatrix} a_k \\ \vdots \\ f_k \end{bmatrix} + v_k \tag{5}$$

Again, in the absence of further information we model the noise sequences $\{v_k\}$ and $\{w_k\}$ as independent zero-mean, white, Gaussian random processes with covariance matrices Q and R respectively.

Whereas the *log-linear form* of the model (2) is rooted in physical modelling, the similar form we have chosen in (4) is done by fiat. We have conducted experiments to validate this assumption. In one experiment, we held the spin speed ω constant and recorded resist thickness t measurements while varying the bake temperature T over a narrow range. In the second experiment, we held the bake temperature constant and recorded PAC concentration measurements while varying the spin speed over a narrow range. In the final experiment, we again held spin speed constant and recorded PAC concentration while varying the bake temperature over a narrow range. We chose to conduct these experiments over narrow ranges to suppress nonlinear effects. Also, we could not use a statistically significant number of trials in our experiments because of significant disturbance effects that cause the model parameters to drift. While the results of our experiments do not conclusively validate the log-linear input-output relationship hypothesized in the model (4), we feel comfortable with our assumption. In large part, this is because over a run of single-product wafers, we would only make incremental changes in input settings to reduce process variation for which an approximate linear model would suffice.

We can write both models (2,3) and (4,5) in the standard form as

$$\begin{aligned} \theta_{k+1} &= \theta_k + v_k \\ y_k &= C_k \theta_k + w_k \end{aligned} \tag{6}$$

Here θ_k is the parameter vector, y_k are the noisy measurements, v_k is the parameter noise, w_k is the measurement noise, and C_k is formed from the input settings of the kth wafer. For example, in the model (4,5),

$$\theta_k = \begin{bmatrix} a_k \\ \vdots \\ f_k \end{bmatrix}, \; y_k = \begin{bmatrix} log(t_k) \\ log(PAC_k) \end{bmatrix}, \; C_k = \begin{bmatrix} \log(\omega_k) & \log(T_k) & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \log(\omega_k) & \log(T_k) & 1 \end{bmatrix}$$

We model the noise sequences $\{v_k\}$ and $\{w_k\}$ as independent zero-mean, white, Gaussian random processes with covariance matrices Q and R respectively. Our experimental procedure for estimating the matrices Q, R is described in Section 4.

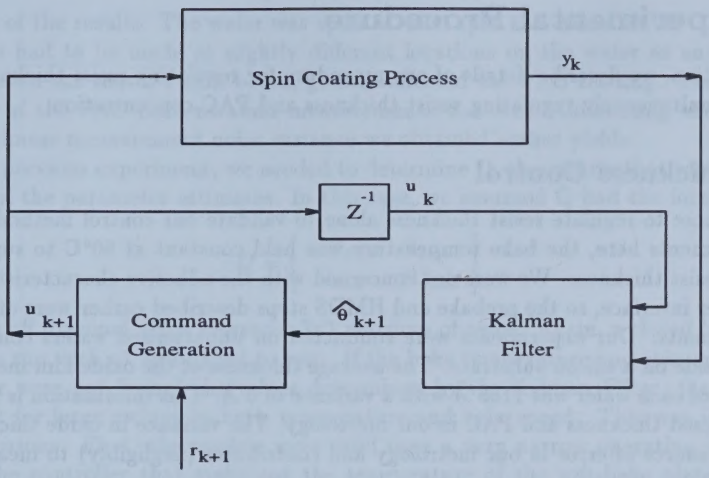


Figure 1: Kalman Filter based adaptive command generation scheme.

3.3 Adaptive Kalman Filter based Control Scheme

The control scheme we will employ to regulate photoresist parameters is illustrated in Figure 1. In order to regulate the $k+1$st wafer, we first obtain an (optimal) estimate $\widehat{\theta_{k+1}}$ of the parameters θ_{k+1} in the model (6). This estimate is based on the past wafer input-output measurements $\{y_i, C_i\}$ for $i = k, k-1, \cdots$ and may be computed recursively by standard Kalman filtering equations (see [10]) as

$$\widehat{\theta_{k+1}} = \widehat{\theta_k} + L_k \left(y_k - C_k \widehat{\theta_k} \right)$$

Here, the Kalman gain L_k is determined by

$$\begin{aligned} L_{k+1} &= P_{k+1} C'_{k+1} \left(C_{k+1} P_{k+1} C'_{k+1} + R \right)^{-1} \\ P_{k+1} &= P_k - L_k C_k P_k + Q \end{aligned}$$

These recursions are initialized by $\widehat{\theta_0}$ which is an *a priori* estimate of the parameters and P_0 which is an estimate of the covariance matrix of our *a priori* estimate $\widehat{\theta_0}$.
Having obtained the parameter estimate $\widehat{\theta_{k+1}}$, we can determine appropriate input settings for the $k+1$st wafer by solving

$$r_{k+1} = C_{k+1} \widehat{\theta_{k+1}}$$

where r_{k+1} are the desired (or reference) output setpoints for the $k+1$st wafer. The only unknown quantities in the above equation are the input settings for the $k+1$st wafer.
In order to implement the control scheme described above, we need to specify the quantities $Q, R, \widehat{\theta_0}, P_0$. Our technique for doing this is described in the next section.

4 Experimental Procedure

In this section, we describe details of our procedure for regulating resist thickness alone, and for simultaneously regulating resist thickness and PAC concentration.

4.1 Thickness Control

We first chose to regulate resist thickness alone to validate our control methodology. In our experiments here, the bake temperature was held constant at 90°C to suppress its effect on resist thickness. We were not concerned with the adhesive characteristics at the resist/wafer interface, so the prebake and HMDS steps described earlier were omitted for all experiments. Our experiments were conducted on unpatterned wafers consisting of silicon dioxide on a silicon substrate. The average thickness of the oxide film measured at the center of each wafer was 1188 $\mathring{A}$ with a variance of 6 $\mathring{A}$. This information is needed to compute resist thickness and PAC in our metrology. The variance in oxide thickness is a secondary source of error in our metrology and contributes (negligibly) to measurement noise.

In this regulation problem we have one actuated input (final spin speed ω) and one sensed output (resist thickness t). In order to determine sensor noise variance R, we coated a wafer with resist and then measured the resist thickness repeatedly at the same position on the wafer. The standard deviation of this thickness measurement was 27 Å. Observe from (2) that the sensor noise variance R reflects uncertainty in the measurements of $\log(t)$ (as opposed to t). The variance of $\log(t)$ can be well approximated by

$$R = E\left[w_k^2\right] = \frac{E[t^2] - E^2[t]}{E^2[t]} \log^2(e) \approx 1.375 * 10^{-6}$$

The procedure to estimate the process noise variance Q is more complicated. To do this, we first acquired an input-output data set on a batch of wafers. In this run, the desired thickness setpoint was held at 13000$\mathring{A}$, and the input spin speed values for the kth wafer were determined based on Kalman filtering estimates of θ_k under the *assumption* of no process noise (i.e. $Q = 0$). The soft-bake time was 60 seconds at a temperature of 90°C. We then hypothesized the structure

$$Q = q \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

for the process covariance matrix and adjusted the single paramter q until the input-output data agreed as closely as possible with the Kalman filter based predicted input-output behavior. Using this procedure we obtain $q = 3 * 10^{-4}$.

4.2 Thickness and PAC control

For the simultaneous regulation of resist thickness and PAC concentration, we used the model (4,5) described earlier. The overall procedure to determine the covariance matrices Q and R here was similar to that used for thickness control alone.

The first step was to determine the sensor noise for the measurement of PAC concentration. We made the reasonable assumption that the measurement noises for resist thickness and PAC were uncorrelated. The PAC measurement noise variance was determined by making repeated measurements of a single wafer and calculating the standard

deviation of the results. The wafer was spun at 4600 rpm and baked at 90°C. The measurements had to be made at slightly different locations on the wafer as any exposure to light, even the sensor's thin beam, greatly affected the PAC reading. The standard deviation of the PAC concentration measurements was .012. Combining this with the resist thickness measurement noise variance we obtained earlier yields
As in the previous experiment, we needed to determine Q, the matrix that represents the variance of the parameter estimates. In this case, we assumed Q had the form,

$$Q = \begin{bmatrix} q1I_{3x3} & 0_{3x3} \\ 0_{3x3} & q2I_{3x3} \end{bmatrix} \tag{7}$$

where the off diagonal zeros represent 3x3 matrices of zeros. Again, a closed loop experiment was run with q1 and q2 set to zero. If the bake temperature and spin speed of the first wafer were not "near" the values determined by the Kalman Filter, the controller would ask for large swings in both temperature and spin speed. This was impractical for two reasons. First, the models were valid over a very narrow operating range, and second, the controller that stabilized the temperature of the soft-bake plate was very slow. A change in temperature of even 10°C could take 15 minutes or longer. This was an unacceptable delay between wafers. For these reasons, we limited the change in temperature that that the controller could ask for to 3°C, and the change in spin speed to 100 rpm. All subsequent experiments were performed in this way with the idea in mind that if a similar setup were actually implemented in production, the temperature of the soft-baking hot plate would be quickly changed and stabilized.
The target thickness value was again 13000 Å, and as always, the target PAC concentration was 1.0. The resulting performance was less than satisfactory. However, these results were used in conjunction with a second Kalman Filter program whose function was to take spin speed and bake temperature measurements and produce the corresponding desired thickness and PAC concentration. q1 was then increased until the desired thickness values matched the measured thickness values, and q2 was increased until the desired PAC concentration values matched those measured. This procedure was identical to our method for obtaining Q in 4.1.

5 Results and Conclusions

To evaluate the efficacy of our control scheme, we have compared the open loop and closed loop performance for resist thickness control alone and for simultaneous control of resist thickness and PAC. These comparisons are shown in Figures 2 and 3 respectively. From our data, we have estimated the open loop variance of the resist thickness to be $\pm 300\mathring{A}$ lot to lot and $\pm 50\mathring{A}$ wafer to wafer. For PAC concentration, the open loop variance was estimated at .02 lot to lot and .023 wafer to wafer.
For control of resist thickness alone, we were able to hold the resist to $\pm 30\mathring{A}$ from the target value, as seen in figure 2. This is clearly a significant improvement over what is possible with open loop operation of a spin coater, or even using batch control. Since we estimated the sensor variance to be 27 Å, our closed-loop resist thickness variance of $30\mathring{A}$ is close to the optimal achievable performance.
For simultaneous regulation of both photoresist properties, the thickness control performance was slightly degraded. This is shown in figure 3. The controller took longer to learn the behavior of the system as there were three parameters for each output as opposed to the two used for thickness control only.

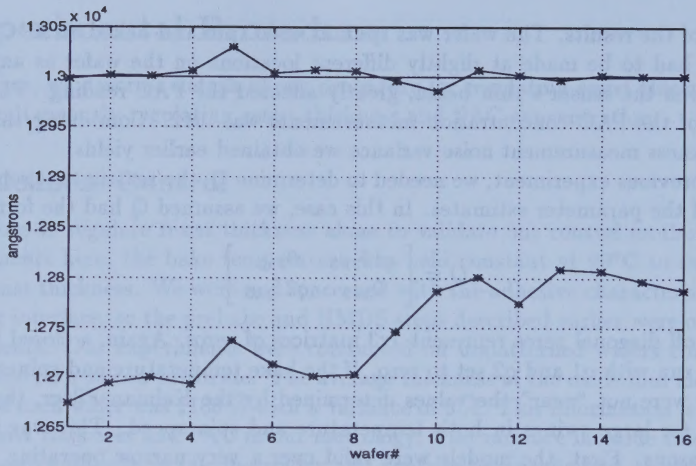


Figure 2: Open Loop / Closed Loop Comparison (Thickness).

In attempting to control the PAC concentration, we recall that the standard deviation of the sensor noise was .012. For the experiment shown in figure 3, both the thickness and PAC concentration were within the limits determined by the sensor noise by the fifth wafer. The standard deviation of the PAC concentration of wafers numbered five through fifteen was .0424, which was actually much worse than the open loop performance. However, when the outlying wafer(#9) was removed from the calculation, the variance diminished to .0138, which was what we would have expected considering the magnitude of the sensor noise. It is our opinion that the conditions exhibited by wafer #9 were caused by the restrictions placed upon the controller by the inadequacies of the hot plate.

Given the fact that the aggressiveness of the controller was limited by the inability of the hot plate to adjust and stabilize quickly, the results of this experiment were encouraging. With only modest capital cost we were able to greatly reduce the variance in photoresist thickness, and moderately reduce PAC variance.

References

[1] D.E. Bornside, C.W. Macosko, and L.E. Scriven, "Spin Coating–One-Dimensional Model," *Journal of Applied Physics*, July 15, 1992, V72 N2:725-740.

[2] R.N. Castellano, *Semiconductor Device Processing: Technology Trends in the VLSI Era*, Philadelphia : Gordon and Breach Science Publishers, 1993.

[3] D.J. Elliot, *Integrated Circuit Fabrication Technology*, New York : McGraw-Hill, 1982.

[4] W.W. Flack, D.S. Soong, A.T. Bell, D.W. Hess, "A Mathematical Model for Spin Coating of Polymer Resists," *Journal of Applied Physics*, August 15, 1984 V56 N4:1199-1206.

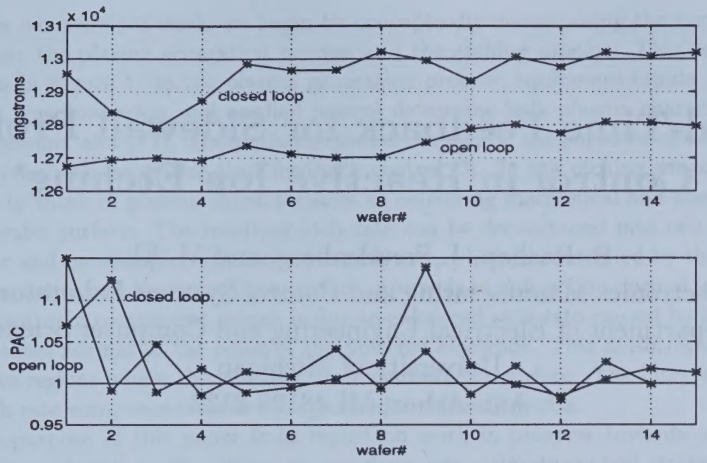


Figure 3: Open Loop / Closed Loop Comparison (Thickness and PAC).

[5] S. Leang, C. Spanos, "A Novel In-Line Automated Metrology for the Photolithography Process," University of California at Berkeley.

[6] S. Leang, Private Communications, University of California at Berkeley, Spring 1994.

[7] L. Ljung, *System Identification Theory for the User*, Englewood Cliffs, N.J. : Prentice-Hall, Inc. 1987.

[8] S. Ma, Private Communications, University of California at Berkeley, Spring 1994.

[9] E. Ng, Private Communications, LSI Logic Corporation, Milpitas, California 95035.

[10] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*, Prentice-Hall, 1982.

[11] W.R. Runyan, K.E. Bean, *Semiconductor Integrated Circuit Processing Technology*, Menlo Park, California : Addison-Wesley Publishing Company, 1990.

[12] D.S. Soane, R.K. Yonkoski, "Model for Spin Coating in Microelectronic Applications," *Journal of Applied Physics*, July 15, 1992, V72 N2:725-740.

[13] C. Spanos, Private Communications, University of California at Berkeley, Spring 1994.

[14] P.C. Sukanek, "A Model for Spin Coating with Topography," *Journal of the Electrochemical Society*, October 1989, V136 N10:3019-3026.

[15] S. Wolf, R.N. Tauber, *Silicon Processing for the VLSI Era, Volume 1-Processing Technology*, Sunset Beach, California : Lattice Press, 1986.

Real-Time Feedback for Sidewall Profile Control in Reactive Ion Etching*

B. Rashap, J. Freudenberg, and M. Elta
Electronics Manufacturing and Control Systems Laboratory
Department of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, MI 48109-2122

Abstract

Reactive ion etching is an important process in the fabrication of microelectronics. This paper reports on work in progress towards developing a strategy for controlling sidewall profile in this process. In this strategy, the plasma characteristics necessary to achieve a desired sidewall profile are first determined. Then, a real-time feedback controller is used to regulate the plasma to these conditions. Presented in this paper are preliminary experimental results towards implementing such a strategy.

1 Introduction

The use of plasmas as dry etching tools has found wide acceptance in the fabrication of VLSI microelectronics. As device dimensions shrink to 0.35 μm and below, tighter control of a number of the fabrication process steps will be needed. Over the past few years, real-time feedback control has been applied to various process steps in an attempt to achieve tighter process specifications. The present paper builds on previous work stabilizing etch rate in a CF_4-based reactive ion etch (RIE) [1, 2, 3]. While stabilization of etch rate shows the utility of real-time multivariable feedback control to reactive ion etching, there are other important etch characteristics that also need to be controlled. These include sidewall profile, selectivity, uniformity, and surface damage. In particular, sidewall profile is important in a number of situations. For example, in the etching of submicron linewidths undercut can not be tolerated, therefore vertical sidewalls are important. In contrast, sloped sidewalls are necessary in areas where step coverage is an issue. This paper presents a strategy for the control of sidewall profile through regulation of plasma characteristics in a RIE process. In this study, we focus on the etching of polysilicon by fluorocarbon plasmas, but the strategy can be generalized to other processes and chemistries.

*This research was supported in part by the Semiconductor Research Corporation under contract no.94-MC-085

As in our previous work, we begin by conceptually decomposing the system into two processes: the plasma generation process and the etching process. This decomposition is shown in Figure 1. In the plasma generation process, equipment inputs (*i.e.* throttle position, gas flow rates, and applied power) determine bulk plasma characteristics (*i.e.* V_{bias}, pressure, and $[F]$). These plasma properties can be measured *in-situ* and therefore can be regulated using real-time feedback control [3]. In the etching process, it is convenient to think of plasma characteristics as delivering mechanical and chemical energy to the wafer surface. The resulting etch rate can be decomposed into two components: isotropic and anisotropic. The isotropic etch rate, which is controlled by the diffusion of reactive particles to the polysilicon surface, proceeds in a direction normal to the surface. The anisotropic component, which is due to enhanced etch rate caused by ion bombardment, etches normal to the plane of the powered electrode. This anisotropic etch rate is absent in regions where the masking layer shadows the surface. The integration of these two etch rate components leads to a particular sidewall profile.

The purpose of this paper is to report on work in progress towards a strategy for controlling sidewall profile. When we are done, our ultimate control strategy (shown in Figure 1) will allow us to achieve a desired sidewall profile through the manipulation of plasma characteristics. This strategy can be broken down into two parts: 1) determining the plasma characteristics necessary to achieve a desired sidewall profile and 2) using a feedback controller to set these characteristics during an etch. The mapping of sidewall profile to plasma properties is, in turn, divided into two parts. First, the etch rate components associated with the sidewall profile are found using a string model simulation, this will be discussed in Section 5. Next, the plasma characteristics necessary to generate these etch rate components are determined using a response surface. Once the necessary plasma properties are found, the etch will be performed using a feedback controller to regulate the plasma to these conditions. In order to implement this strategy, we must first develop a linear model of the plasma generation process. From this model, a real-time feedback controller can be designed to hold desired plasma characteristics despite disturbances. Next, for a variety of plasma conditions the resulting sidewall profile will be mapped into etch rate components. Results from different operating points will be used to construct a response surface relating plasma characteristics to etch rate components. In this paper, we report successful completion of these steps for a single operating point. As described in the final section, these steps remain to be repeated for various operating conditions so that the response surface may be developed.

2 Experimental Apparatus

This work has been performed using an Applied Materials 8300 RIE. The experimental setup is described in detail in [1]; for completeness, a brief description of the reactor is presented here. The Applied Materials 8300 has a hexagonally shaped powered electrode with the capability for simultaneously etching eighteen 4 inch wafers; however, in this work it is used as a single wafer etcher. Power is supplied by a 13.56 MHz rf generator and impedance matching is through an RF Services RFS8300RL matching network. Several MKS 1259C Mass Flow Controllers regulate process gas flows; the gases are mixed in a manifold before entering the chamber. An MKS 652A Throttle Valve and a 653A Throttle Valve Controller are used regulate pumping speed of a turbomolecular exhaust pump. It is standard practice in process control to allow the throttle valve controller

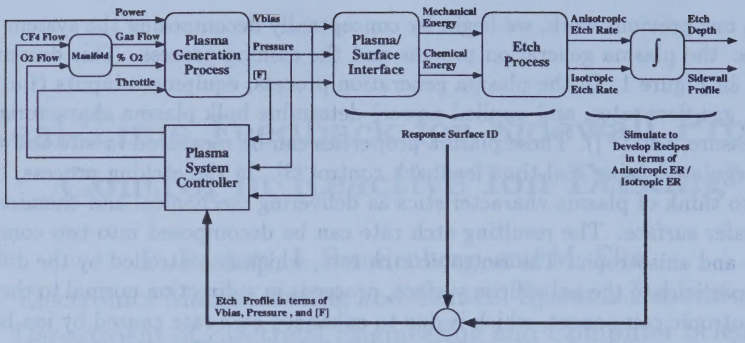


Figure 1: Sidewall Control Strategy

to automatically manipulate the valve to achieve a desired pressure setpoint. However, because we wish to develop a multivariable relationship between equipment inputs and plasma properties, we control the throttle valve position directly.

There are three measurements of plasma characteristics that are regulated in the control strategy: the bias voltage (V_{bias}) between the powered and grounded electrodes, the chamber pressure, and the concentration of fluorine ($[F]$), which is the dominant reactive species. Bias voltage is measured through an inductive tap at the chamber side of the matching network. Chamber pressure is monitored by an MKS Baratron Capacitance Manometer with a range from 1 - 100 mTorr. Fluorine concentration is inferred from optical emission spectroscopy using actinometry [4] with argon as the calibration species. Details of the actinometry system have been presented in [1, 3]. This system has been recently upgraded to include two SPEX 500M half-meter monochromators with thermoelectrically cooled Hamamatsu R928 photomultiplier tubes. This has led to increased resolution and repeatability in measuring the intensities of wavelengths in the fluorine and argon spectra.

3 Process Chemistry

Our first application of feedback control to reactive ion etching [1] involved the use of a single process gas, CF_4. The initial goal was to control three plasma characteristics (bias voltage, pressure, and fluorine concentration) by manipulating three equipment input parameters (throttle position of the exhaust conductance controller, flow rate of the process gas, and applied rf power). It turned out that in the operating region for our process, throttle position and flow rate had similar effects on the plasma characteristics and thus could not be used as independent actuators. Further analysis revealed that throttle position had more control authority. Hence we could use throttle position and applied power as actuators to control any two of the three plasma characteristics [2].

In using plasma characteristics to control sidewall profiles, it is likely that all three of the characteristics mentioned above play an important role. Therefore, in order to obtain independent control of the plasma characteristics, O_2 is added to the CF_4 chemistry and $\%O_2$ is used as a third regulated input while the total gas flow rate is held constant. In order to use optical emission spectroscopy to measure fluorine concentration argon

is added as 5% of the overall gas flow; hence, the percentage of oxygen is actually the percentage of the remaining 95% of the flow. The addition of O_2 to the plasma liberates atomic fluorine and forms CO_x through interactions with fluorocarbon molecules. This leads to enhanced fluorine concentrations and prevents polymers from forming on the polysilicon surface (this will be discussed subsequently). Because there is only a small percentage of dissociation of the feed gases and the percentage of O_2 is also kept small, the addition of oxygen has little effect on the bias voltage and the chamber pressure. Thus, this slightly more complicated process chemistry allows independent actuation of all three plasma characteristics.

In general, sidewall passivation is used to minimize undercut during the etch process. In a CF_4 chemistry this passivation occurs when polymers form on the surface in areas where ion bombardment is not present [5]. In order to focus our research, we have decided to simplify the plasma/surface interactions by not exploiting sidewall passivation. While the addition of hydrogen into the plasma chemistry allows the potential for control of polymerization, there is a significant amount of preliminary research that needs to be performed before it could be implemented in a real-time control strategy. This is beyond the scope of the current research, but will be pursued at the University of Michigan in the future.

4 Control of the Plasma Generation Process

The control strategy employed to regulate the three plasma characteristics was similar to that used in [3]. First the dynamic responses of the RIE to step changes in the inputs were obtained. Next, we used these responses to develop an empirical model of the local process dynamics. Finally, we designed a control algorithm using the model and applied it to regulate the plasma characteristics to desired setpoints.

Empirical modeling of the plasma generation process was performed by first selecting an operating point around which the linear model was developed. Several factors were important in choosing this nominal operating condition. First, the reactor needed to be operating in an RIE mode, where both physical and chemical etch mechanisms are present. It was also important that none of the actuators or sensors operated in a region too close to saturation. While standard practice is to operate the plasma under conditions where its properties are relatively insensitive to actuator variations, the opposite is more desirable under feedback . Therefore, operating points were chosen to yield adequate actuator authority. Finally, even though it is well known that plasmas are nonlinear, the models and controllers used in this study were linear. Therefore, an attempt was made to find a region where the dynamics of the plasma generation process are approximately linear. These requirements led to an operating point, in terms of equipment inputs, of a throttle 15% open, 5% O_2 (with a total $CF_4/O_2/Ar$ flow rate of 30 sccm), and an applied power of $1000W$.

Data to be used in the empirical modeling was collected in the following manner. A 20% step increment in throttle position was applied to the system while the other actuators were held constant. This experiment was repeated 10 times and the results averaged together to improve the signal-to-noise ratio of the fluorine measurement while retaining the time response of the signal. This procedure was repeated for both the $\%O_2$ and power inputs. The data was then visually inspected to determine any pure time delays in the response of the plasma to changes in the inputs. Next, we used

standard system identification algorithms [6, 7] to develop a transfer function matrix that related small signal deviations in the RIE inputs to changes in the plasma characteristics. Finally, each of the inputs and outputs from the plasma generation process was scaled by its nominal values as a way of normalizing the system to unit deviations in each. The resulting model of the system was give by:

$$\begin{bmatrix} V_{bias} \\ Pressure \\ Fluorine \end{bmatrix} = \begin{bmatrix} \frac{1.65e^{-.33s}}{s+0.16} & \frac{0.25e^{-.77s}}{s+0.41} & \frac{12.23(s+0.27)}{(s+0.19)(s+62.42)} \\ \frac{-0.97e^{-.33s}}{(s+0.18)(s+3.00)} & \frac{0.024e^{-.77s}}{s+0.40} & \frac{-0.011(s-0.006)}{(s+0.19)(s+2.33)} \\ \frac{4.85(s-0.73)e^{-.33s}}{(s+0.11)(s+39.76)} & \frac{0.33e^{-.77s}}{s+0.17} & \frac{0.49(s+0.067)}{(s+0.095)(s+19.69)} \end{bmatrix} \begin{bmatrix} Throttle\ Position \\ Percent\ O_2 \\ Applied\ Power \end{bmatrix}$$

Based on this model, we designed a feedback controller for the plasma generation process which satisfied a number of requirements. First of all, it was important that the controller be able to drive the system to constant setpoints of the plasma characteristics. In addition, because of the nonlinear nature of the throttle valve and the possibility of saturation of the $\%O_2$ input, overshoot in the actuator responses was to be kept to a minimum. Finally, the system needed to reject sensor noise, particularly from the fluorine measurement. After augmenting the model with integrators, in order to achieve zero steady error, the LQG/LTR design technique [8] was used to design the controller. This design was iterated several times to achieve acceptable transient response times, actuator overshoots, and noise rejection.

5 Etch Study

There are no methods available to measure sidewall profile *in-situ* and in real time. For this reason, an etch study is performed to build a response surface from constant plasma characteristics to sidewall profile. These profiles are then analyzed to yield the associated etch rate components and thus allow the construction of a model relating plasma properties to isotropic and anisotropic etch rates. In this section, we will present the steps necessary to obtain this relation for one operating point and then, in the final section, discuss how the results from several points can be used in our strategy to control sidewall profile.

In this etch study, the real-time feedback controller is used to control the RIE at constant plasma characteristics during the etching of a $PolySi/SiO_2/Si$ wafer. The selectivity of polysilicon to photoresist in a CF_4/O_2 plasma is very poor. Once again, to focus the research, we use a nickel mask which will not etch in a CF_4 plasma. This allows us to concentrate on modeling the effect that plasma properties have on etch rate components without having to account for the effects of mask erosion. A lift-off process [9] is used to define the nickel mask, which contains 1 μm, 2 μm, and 5 μm linewidths of various spacing.

As an initial starting point for the etch study, the plasma conditions associated with the nominal operating point for the system identification were used. These turned out to be a bias voltage of 360 V, a chamber pressure of 18.22 mTorr, and a fluorine concentration of 10.08 (arbitrary units). As mentioned in [3], the Applied 8300 RIE does not have a load lock. Therefore, outgassing of moisture from the chamber walls during the first part of the etch causes a large perturbation to the plasma characteristics. As can be seen

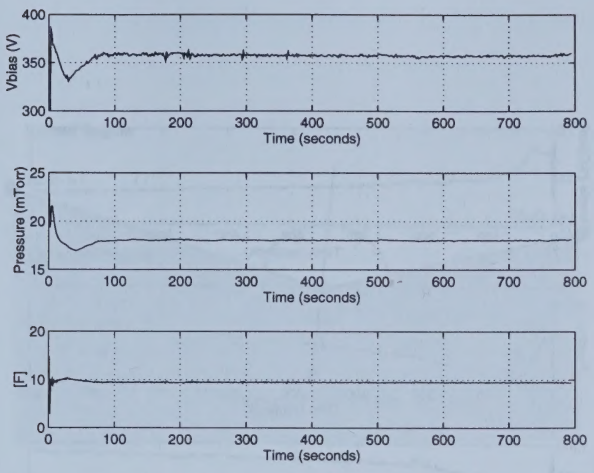


Figure 2: Plasma characteristics during feedback control of the plasma generation process.

in Figure 2, the controller compensates for this disturbance and drives the plasma characteristics to their desired setpoints. The corresponding actuator commands are shown in Figure 3. After the etch, the wafer is diced and a scanning electron microscope (SEM) photograph is taken of the sidewall profile (Figure 4).

The final step is to extract the isotropic and anisotropic etch rate components from the SEM image. This is done by using a string model [10] to obtain a simulated sidewall profile from specified etch rate components. In the string model, the exposed polysilicon surface is represented by a series of nodes connected by straight line segments. The progression of each node is determined by its local etch rate. This etch rate is the vector sum of an isotropic component along the angle bisector of the two adjacent segments (see Figure 5) and an anisotropic component in regions not masked by the nickel. As the separation between points changes during the etch simulation, nodes are added and subtracted from the string to maintain sufficient resolution and relieve computational load. This model has been implemented in C and uses etch rate components and etch time to produce a profile.

The string model simulation is iterated to produce a match with the sidewall profile obtained using the SEM. The SEM photograph is scanned into digital format and processed using an edge detection algorithm from the MATLAB Image Processing Toolbox [11] to extract the sidewall profile. As can be seen in Figure 6, the simulation does an excellent job of capturing slope and shape of the sidewall profile. There is some mismatch between the two in the corners, likely due to the simulation not accounting for local diffusion effects of the reactive species. For this operating point, the etch rate components were found to be 6.0 Å/s isotropic and 6.5 Å/s anisotropic.

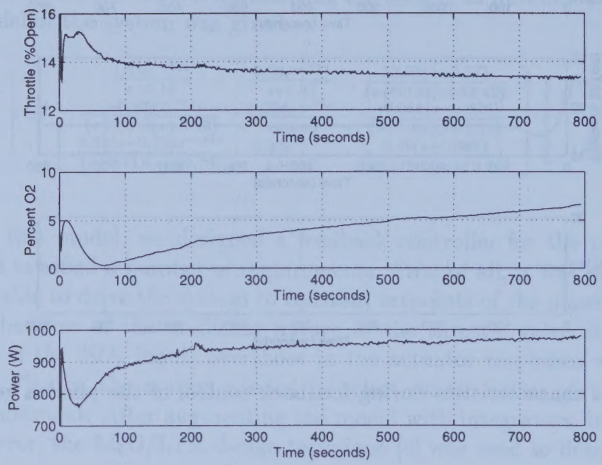


Figure 3: Corresponding actuator responses during feedback control.

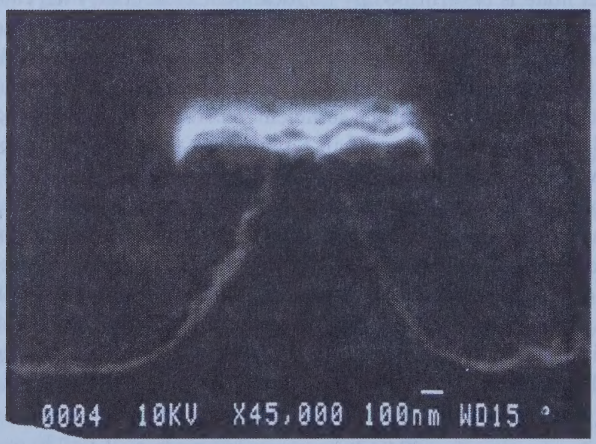


Figure 4: SEM image of resulting sidewall profile.

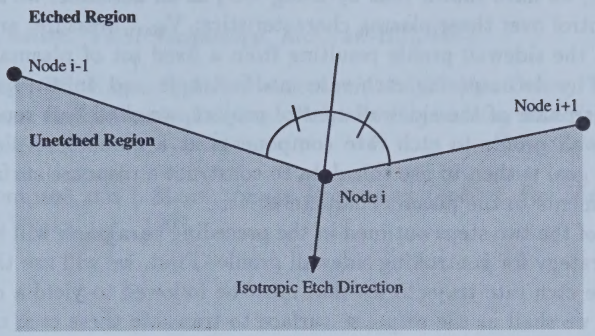


Figure 5: Direction of isotropic etch rate

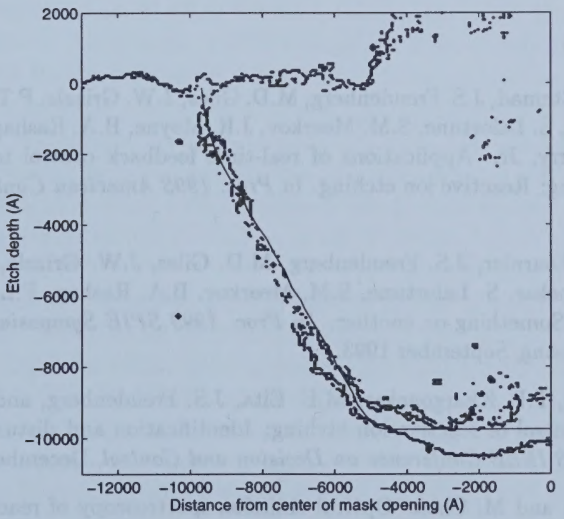


Figure 6: Comparison of extracted edge profile (dots) with results from string simulation (solid).

6 Conclusions and Further Work

In this paper, we have extended the RIE control work described in [1, 2, 3] in two directions. First, we have shown that by using $\%O_2$ as an actuator, we are able to gain independent control over three plasma characteristics: V_{bias}, pressure, and $[F]$. Second, we showed that the sidewall profile resulting from a fixed set of plasma characteristics my be modeled by decomposing etch rate into isotropic and anisotropic components. To complete this phase of the sidewall control project, we shall first repeat the process of relating sidewall profile to etch rate components at a number of plasma operating conditions. Our goal is then to use this data to construct a response surface relating the etch rate components to the plasma characteristics.

Completion of the two steps outlined in the preceding paragraph will enable us to use the following strategy for controlling sidewall profile. First, we will use the string model to determine the etch rate trajectories that must be followed to yield a desired sidewall profile. Second, we shall us the response surface to translate these etch rate trajectories into corresponding trajectories that must be followed by the plasma characteristics. Finally, the real-time controller will be used to force the plasma characteristics to follow these desired trajectories.

The implementation of this control strategy will provide a mechanism for translating desired sidewall profiles into specific etch recipes. Once it is shown that this is possible, the work will be extended to more complicated etch situations. This will include using a more industrially relevant masking layer, such as photoresist, and working with an etch process which takes advantage of sidewall passivation.

References

[1] M. Elta, H. Etemad, J.S. Freudenberg, M.D. Giles, J.W. Grizzle, P.T. Kabamba, P.P. Khargonekar, S. Lafortune, S.M. Meerkov, J.R. Moyne, B.A. Rashap, D. Teneketzis, and F.L. Terry, Jr. Applications of real-time feedback control to semiconductor manufacturing: Reactive ion etching. In *Proc. 1993 American Controls Conference*, June 1993.

[2] M. Elta, J. Fournier, J.S. Freudenberg, M.D. Giles, J.W. Grizzle, P.T. Kabamba, P.P. Khargonekar, S. Lafortune, S.M. Meerkov, B.A. Rashap, F.L. Terry, Jr., and T. Vincent. Something or another. In *Proc. 1993 SPIE Symposium on Microelectronic Processing*, September 1993.

[3] B.A. Rashap, P.P. Khargonekar, M.E. Elta, J.S. Freudenberg, and F.L. Terry, Jr. Real-time control of reactive ion etching: Identification and disturbance rejection. In *Proc. 1993 IEEE Conference on Decision and Control*, December 1993.

[4] J.W. Coburn and M. Chen. Optical emission spectroscopy of reactive plasmas: A method for correlating emission intensities to reactive particle density. *J. Appl. Phys.*, 51(6):3134, June 1980.

[5] J.W. Coburn and H.F. Winters. Plasma etching - a discussion of mechanisms. *J. Vac. Sci. Technol.*, 16(2):391–403, Mar./Apr. 1979.

[6] L. Ljung. *System Identification: Theory for the User.* Prentice-Hall, Inc., 1987.

[7] T. Soderstrom and P. Stoica. *System Identification.* Prentice Hall, Inc., 1989.

[8] J.C. Doyle and G. Stein. Multivariable feedback design: Concepts for classical/modern synthesis. *IEEE Trans. Automatic Control*, 26(1):4–16, February 1981.

[9] S.M. Sze, editor. *VLSI Technology.* McGraw-Hill, 1983.

[10] R.E. Jewett, P.I. Hagouel, A.R. Neureuther, and T. Van Duzer. Line-profile resist development simulation techniques. *Polymer Engineering and Science*, 17(6):381–384, June 1977.

[11] C.M. Thompson and L Shure. *Image Processing Toolbox: For Use with MATLAB.* The MathWorks, Inc., 1993.

Control Analysis and Design for Czochralski Growth of GaAs[1]

Douglas P. Looze[2], F. Azad[3], B. Bernstein[3], and T. Collins[3]

Abstract

This paper describes the development and testing of a model-based digital control system for the growth of bulk Gallium Arsenide using the liquid encapsulated Czochralski technique. The resulting control system provides more uniform crystal growth than current commercial systems, and is readily scaled to larger processes.

1. Introduction

Gallium Arsenide (GaAs) Integrated Circuits (ICs) switch at faster rates, pass signals faster, exhibit lower noise, consume less power at high speeds (up to four times the speed at less than one half the power), operate over a wider temperature range and are inherently more radiation hard than silicon ICs. Gallium Arsenide has already found significant military and commercial application in light emitting diodes, monolithic microwave integrated circuits (MMICs), analog integrated circuits and hybrid devices. In addition, increasing use in Digital IC (DIC) applications is foreseen in the near future. Most of the growth in the use of GaAs will be in commercial high speed computer and high frequency communications applications, rather than in military applications as in the past.

Growth in GaAs applications has been hindered by difficulties in the manufacturing process. One of these difficulties is efficiently obtaining crystals from which defect-free wafers four inches or greater in diameter can be cut. Semi-insulating GaAs crystals (used for MMICs, hybrids, and DICs), like silicon crystals, are grown by the Czochralski (Cz) process. However, unlike silicon, GaAs is a compound of two elements, one of which (arsenic) is volatile at the crystal melting point. Consequently, a modified process referred to as the Liquid-Encapsulated Czochralski (LEC) process must be used in which a layer of boron oxide is maintained over the molten GaAs to prevent arsenic vaporization. Control of the LEC process for GaAs is intrinsically more difficult than control of the standard Cz process for silicon and, in addition, much greater development effort has been expended for the latter as compared to the former process. It has been estimated that the technology gap between the two processes is about 20 years.

Intelligent control of the LEC process provides the potential for numerous improvements both at the diagnostic and supervisory control level and at the real time control level. Improved diagnostic control could reduce material waste caused by formation of polycrystalline structures and reduce requirements for human monitoring of the process. Real time control could both improve process efficiency as well as increase the quality of the wafers produced. However, control of the Cz process is complicated greatly by the anomalous (nonminimum phase) behavior of the available process measurement (the crystal weight).

The use of an automatic control system can provide many benefits for the LEC growth of GaAs crystal. A good control system can grow boules with nearly uniform radius

[1] This work has been supported by DARPA through ARO under contract sDAAL03-86-C-0022 and DAAL03-91-C-0009.

[2] Dept. Electrical and Computer Engrg., University of Massachusetts, Amherst, MA, 01002

[3] The General Electric Company, Corporate Research and Development, Schenectady, NY 12301

and can minimize the variations in growth rate. A process that can grow crystals with more uniform boule radius can provide more wafers from a given melt size by minimizing the material wastage. A more uniform growth environment may translate into crystal with lower dislocation densities by reducing the thermal stresses due to growth rate variations, and may delay the onset of twinning or polycrystalline growth. A reliable control system can also reduce the need for operator attention, thereby reducing the operator workload. Each of these potential benefits increases the efficiency of the process.

Weighing control techniques for diameter control of Gallium Arsenide grown with the liquid encapsulated Czochralski (LEC) method have been long studied (c.f., [1-10]) and are currently implemented on high pressure pullers growing 8 kg boules of 3 and 4 inch diameter. However, real-time control of the Cz process is complicated greatly by the anomalous (nonminimum phase) behavior of the most crystal weight measurement. Currently implemented commercial control systems handle the effects of the system dynamics heuristically. The development of more effective control algorithms that achieve better closed-loop performance and can be readily scaled to larger processes requires the development of models of the growth process dynamics, and the use of these models in the design of the control algorithm.

This paper presents the design and analysis of a model-based digital control system for the GaAs LEC process. The digital control system incorporates all the logic required from the initial growth of the boule to the termination of the growth process, as well as the ability to handle operator-directed halts and restarts of the growth process. The compensation algorithm used for closed-loop growth was developed based on analytic models that have been verified by experimental identification. The resulting control system was able to grow an 8 kg, 4 inch boule with superior shape.

2. Dynamic Model of the Liquid Encapsulated Czochralski Process

2.1 Overview

Semi-insulating GaAs is commonly grown as a single crystal via the liquid encapsulated Czochralski (LEC) technique. The two variations of the LEC procedure that are currently employed commercially for GaAs crystal growth differ primarily in the pressure used during synthesis of the GaAs compound and growth of the crystal. The low pressure technique typically employs 2-3 atmospheres of pressure during the crystal growth, while the high pressure technique grows crystals under 20 atmospheres of pressure. The system utilized for this study is the high pressure process.

The LEC growth process is a slight variation of the widely employed Czochralski (Cz) technique (c.f., [11-12]). The Cz technique is quite simple in concept (see Fig. 1). A pool of molten material (the melt) is maintained at a temperature slightly above the melting point. A single crystal "seed" is dipped into the surface of the melt, causing heat to flow from the liquid, and the liquid to begin solidifying on the end of the seed. The single crystalline structure of the seed is replicated by the molecules joining the growing solid. The solid material is withdrawn from the melt at a rate that maintains a solid-liquid interface near the surface of the melt. The pull rate of the crystal and the heat supplied to the melt must be carefully balanced to maintain thermal equilibrium and a steady growth rate with a constant

desired radius. Due to the volatility of arsenic at the melt temperature of GaAs, the basic Cz method requires modification. A layer of molten boric oxide (B_2O_3) is floated on the GaAs melt during synthesis and growth to inhibit the arsenic vaporization. Although the basic Cz process is the same, the B_2O_3 layer has a slight effect on the thermodynamics of the process and a significant effect on the measurement of process dynamics.

2.2 Process Dynamics

A schematic of the LEC process is shown in Fig. 1. The basic process dynamics are governed by the thermodynamics of the heater, the crucible, the melt, the solid boule, the encapsulant, and the argon atmosphere, and by the capillary dynamics of the liquid-solid interface. The heater is typically controlled by a PID control loop with an overall time constant of approximately 100 seconds. The temperature of the melt is maintained by supplying a temperature command to the controlled heater. Heat flows from the heater to warm the crucible walls and the melt. The melt transfers heat from crucible to the liquid solid interface both by convection and by conduction. Heat also flows from the melt to the encapsulant, and from the encapsulant to the argon atmosphere. Heat is extracted from the liquid-solid interface primarily through the solid boule. The boule transfers heat to the encapsulant and to the atmosphere above the encapsulant.

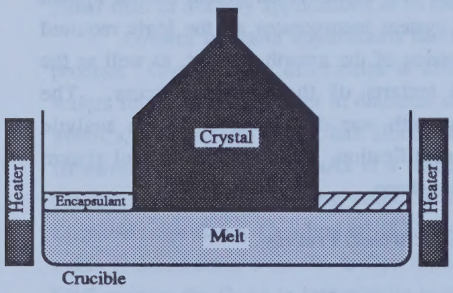


Figure. 1. Schematic of the LEC process.

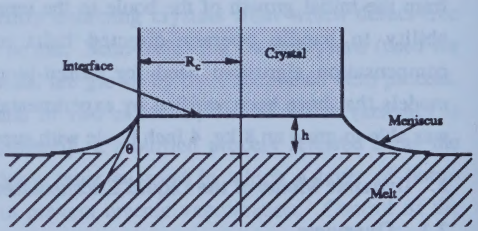


Figure 2. Schematic of the meniscus region.

A precise thermodynamic model of the LEC process requires accurate dynamic modeling of the temperature gradients at the solid-liquid interface, which in turn requires accurate dynamic modeling of the convection and conduction processes in the melt, encapsulant and argon atmosphere. Such a model, if available, would be far too complex for process control design employing current control technology. Low order lumped parameter models appear to provide sufficient accuracy for the purposes of radius control ([7-9]). Such models represent heater, melt, submerged solid, and the encapsulant by their aggregate temperatures, resulting in one thermal lag for each element.

The growth of the crystal is governed by the thermodynamics of the liquid-solid interface, and by the capillary dynamics of the meniscus (Fig. 2). The growth rate of the crystal is determined by the heat transfer across the interface, which in turn is governed by the temperature gradients in the liquid and solid, and by the heat of fusion of GaAs. The temperature gradients in the liquid are affected by the shape of the meniscus, which can be modeled by the Laplace-Young capillary equations.

Significant simplification of the interface dynamics model can be achieved by assuming that the interface is flat. The Laplace-Young equations can be solved for the complete interface shape, which can be used to determine the volume of the meniscus V_{men} and the contact angle θ of the meniscus with solid. The volume is determined by the crystal radius R_s, the meniscus height h, the contact angle, and the capillary constant a (which depends only on the material properties of GaAs):

$$V_{men} = \pi R_s^2 h + \pi a^2 R_s \cos(\theta) \tag{1}$$

The Boucher-Jones approximation provides an accurate expression of the contact angle in terms of meniscus height and crystal radius:

$$h = a\sqrt{\frac{1-\sin(\theta)}{1+\frac{a}{R_s\sqrt{2}}}} \qquad \theta = \sin^{-1}\left(1-\left(\frac{h}{a}\right)^2\left(1+\frac{a}{R_s\sqrt{2}}\right)\right) \tag{2}$$

Conservation of energy across the liquid-solid interface determines the axial growth rate V_g. The flat interface assumption allows the growth rate to be expressed as a function of the temperatures and volumes of the submerged solid and meniscus. With the meniscus volume given by (1) and the solid volume approximately constant, the growth rate can be written as a function of the temperature of the submerged solid T_s, the temperature of the melt T_m, the crystal radius, and the meniscus height:

$$V_g = V_g(T_s, T_m, R_s, h) \tag{3}$$

The rate of change of the meniscus height is given by the geometric analysis in terms of the pull rate V_p, the crucible lift rate V_c, the crystal growth rate, and the rate of change of the melt height h_m:

$$\dot{h} + \dot{h}_m = -V_g + V_p - V_c \tag{4}$$

The radial growth (rate of change of the radius R_S) of the crystal is determined by the contact angle θ between the meniscus and the solid by the equation:

$$\dot{R}_s = V_g \tan(\theta - \theta_0) \tag{5}$$

where θ_0 is the contact angle required to maintain constant radius growth due to the incomplete wetting of GaAs.

The thermal environment of the LEC process is represented by four thermal masses corresponding to the heater (temperature T_h), the melt and meniscus (temperature T_m), the submerged solid (temperature T_s), and the encapsulant (temperature T_e), and a sink corresponding to the ambient environment. The heater is assumed to be tightly regulated by a PID controller with a closed-loop time constant τ of 100 seconds. Thus, the dynamics of the heater temperature T_h is adequately represented by a first order lag:

$$\dot{T}_h = -\frac{1}{\tau}T_h + \frac{1}{\tau}T_{hc} \tag{6}$$

The heat flows between the masses of the melt, submerged solid crystal, encapsulant and ambient are represented by first order dynamics using effective heat transfer coefficients. The general form of the energy balance for component i of the lumped-parameter system is given by:

$$m_i c_i \dot{T}_i = \sum_{j \neq i} A_{ij} h_{ij} \left(T_j - T_i \right) \tag{7}$$

The values of the heat transfer coefficients are estimated by extracting lump temperatures and heat fluxes from a finite element model of the LEC process [13].

The boule weight is given by (c.f., [6]):

$$W = m_s - \rho_e V_{sub} + \left(\rho_m - \rho_e \right) V_{men} \tag{8}$$

Hence, the differential boule weight is given by:

$$\dot{W} = \pi R_s^2 \rho_s V_g - \rho_e \dot{V}_{sub} + \left(\rho_m - \rho_e \right) \dot{V}_{men} \tag{9}$$

Equations (1)-(5) represent a second order, nonlinear differential equation that describes the aggregate interface dynamics in terms of the radial growth rate and meniscus height. Additional dynamics describe the change in the melt level and the submerged crystal area. When combined with the lumped parameter model (6)-(7) a low order (9 state) model of the process suitable for control design is obtained. The input variable for the model is the commanded heater temperature, while the measurement variable is the differential weight (9).

2.3 Control Objectives

Intelligent process control offers numerous potential improvements in the efficiency and quality of the LEC GaAs process. The two primary goals of control for the LEC GaAs growth process are to produce material with better improved properties and to increase the efficiency of the process. Although the relationship between the material properties and the macroscopic growth process is not fully understood, several mechanisms for obtaining improved electrical properties have been proposed. In particular, it has been postulated that the incidence of dislocations in the GaAs crystal is related to thermal stresses of growth. A theoretical thermal stress model developed in [11] has produced a prediction of dislocation densities in GaAs that correlates well with observed data. Although it is impossible to eliminate thermal shear stress during the growth process, variations in the stress can be minimized by maintaining uniform thermal gradients in the solid, thereby maintaining a (presumably) acceptable growth environment. Uniform thermal gradients are reflected in the process as uniform crystal growth. Thus, one control system objective is to maintain a constant crystal radius and a constant axial growth rate. Also, the results of [11] indicate that smaller thermal gradients across the liquid-solid interface result in reduced dislocation densities. However, such reduced thermal gradients lead to an inherent shape instability, requiring even more effective control of crystal radius and growth rate.

Although the radius and axial growth rate of the crystal are not available directly as measurements of the process, the differential weight gain signal will be zero only if the preceding variables are constant. Thus, the objective of the control system design will be to control the heater temperature to minimize variations in the differential weight gain of the crystal.

3. Linearized Analysis of the Model

Because of the stringent requirements on the control system during the transition from shoulder to body growth, the model will be analyzed at the stage of growth just past the shoulder. The boule is assumed to have a shoulder of 70 mm and a current length of 130

mm. The nominal values of the state variables and control variables for this operating point are shown in Table 1, and the parameter values for this operating point are shown in Table 2. For these values, the time derivatives of the lumped-element temperatures, the meniscus height, and the crystal radius are essentially zero.

The remaining variables (the melt height, the submerged area, the submerged volume, and the crystal length) do not have a zero derivative at this operating point, but vary slowly as the crystal geometry changes. The time scale of this variation and the weakness of the coupling to the dominant dynamics allow these variables to be treated separately (c.f. [16]).

The linearized system matrices were computed using a numerical differentiation algorithm. The dominant mode of the linearized model is governed by a pair of lightly damped, complex eigenvalues at $-6.52\times10^{-5}\pm$ j6.02×10^{-4}. This mode corresponds to a swing between radial growth and thermal balance at the interface. The dominant zeros in the system are a nonminimum phase zero at 1.73×10^{-3} (corresponding to the buoyancy in differential weight measurement) and a real zero at -4.81×10^{-4}. Four modes corresponding to the thermal lags of the system are governed by real eigenvalues at -5.05×10^{-3}, -1.00×10^{-2}, -1.79×10^{-2}, and -5.92×10^{-2}. Two zeros resulting from the thermal network are located at -3.10×10^{-2} and -5.97×10^{-3}. A Bode plot of the transfer function is shown in Fig. 3.

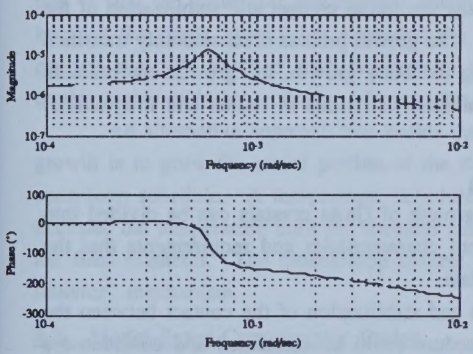


Figure 3. Bode plots of the linearized process model.

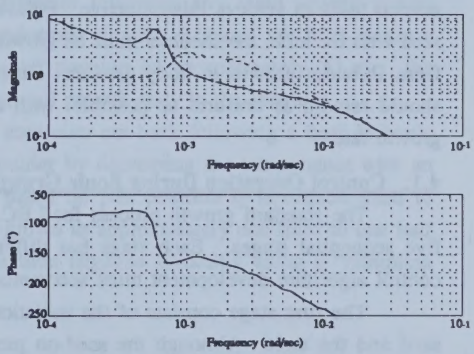


Figure 4. Bode plots of the loop transfer function of the compensated process.

Both the pole-zero analysis and the frequency domain analysis reveal a lightly damped mode with a natural frequency of approximately 6×10^{-4} rad/sec. If excited, this mode would result in undesirable axial growth rate and radius oscillations. The control bandwidth of the system (i.e., the gain crossover frequency of the compensated open-loop transfer function) would have to be somewhat greater than the natural frequency of the mode.

The presence of a right half-plane zero at 1.7×10^{-3} rad/sec limits the control bandwidth that can be achieved with the weight control methods [14]. Specifically, the control bandwidth cannot be much greater than this frequency without significantly amplifying thermal disturbances and degrading the growth process. The effect of the right half-plane zero can be seen in the Bode plots (Fig. 3). The phase begins to decrease significantly at approximately 10^{-3} rad/sec without a commensurate decrease in transfer

function magnitude. To achieve a bandwidth greater than 10^{-3} rad/sec, a controller must to supply phase lead near the gain crossover frequency. The addition of phase lead requires an increase in the slope of the magnitude. However, any significant increase in the slope of the magnitude would preclude having a gain crossover frequency until the plant itself begins to roll off. Thus, any crossover frequency greater than 2×10^{-3} rad/sec would result in a closed-loop system with small phase margins and correspondingly large peaks in the closed-loop transfer functions.

The dominant features of the linearized model have been confirmed through identification experiments on a high pressure puller [13]. Specifically, the location of the oscillatory modes and the high frequency phase behavior of the process and the model agree to within a few percent. The only significant deviation between the model and the process is that the overall gain of the process is approximately 60% greater than that predicted by the model. Although the source of this difference has not been identified, the increased gain can be readily handled in the design process.

4. Digital Control Design and Implementation

The primary objective of the control system operation is to efficiently grow a GaAs boule with good electrical properties and uniform shape. The control system must perform several tasks to achieve this objective. Following the initial contact and stabilization of the seed with the melt, the shoulder must be grown. The growth process must smoothly transition from shoulder growth to body growth. During the body growth, the growth environment should remain as uniform as possible, with minimal variations in radius and longitudinal growth rate.

4.1. Control Operation During Boule Growth

The standard growth process for LEC growth of GaAs crystals can be divided into five sequential stages. Each stage has different characteristics and requirements that the control algorithm development must accommodate.

The first stage consists of the initiation and stabilization of the contact between the seed and the melt. Although the seed-on procedure could be automated and considered a part of the control system, for the purposes of this project it is assumed that it is conducted manually (as with the CI358 controller). Thus, during this stage, the control system is idle, and generates no (zero) control signal. However, other monitoring functions (such as the generation of the differential weight and error signals) are performed.

The second stage is the initial shoulder growth. This stage is characterized by a poor differential weight signal (due to the small boule radius), relatively unstable growth dynamics reflected in the characteristic flash (rapid radial growth) of the boule, and poor models of the growth dynamics. The control objectives of this stage are simply to grow the radius outward toward the shoulder. The rate of growth and the radius profile are only limited by how they affect the ability of the controller to stabilize the process in later stages. The limited control objectives can be accomplished by open-loop control.

The third stage of growth is the latter portion of the shoulder growth and the transition to body growth. The shape of the boule in this stage is important, and the control system objectives are therefore more stringent. The differential weight signal and the models

of dynamic growth are relatively good. The control system for this stage employs a closed-loop algorithm that uses the differential weight signal to generate the temperature command. The control parameters and gain are selected to achieve a small error between the desired and actual differential weight signals.

The fourth stage is the growth of the crystal body. This stage is characterized by a relatively good differential weight signal and dynamic model. Although it is important to maintain nearly constant radius and longitudinal growth, the disturbances that cause these signals to deviate from their desired values tend to be relatively small. Thus, the overall control system requirements are less stringent than during the preceding stage. A closed-loop control algorithm is again implemented, with lower gain parameters to improve the disturbance rejection properties.

The fifth stage is the growth of the tail cone of the boule. This stage is implemented to provide a smooth transition to the end of growth, and is relatively unimportant. The tail cone is generated by supplying a decreasing reference to the closed-loop algorithm.

4.2. Initial Shoulder Growth via Open-Loop Temperature Control

One method for initial shoulder growth is to use closed-loop control with an increasing differential weight reference to grow the entire boule shoulder. A consequence of this is that the controller reduces the boule diameter following any rapid growth (flash) because the boule growth has exceeded the reference signal. Because the high-pressure process typically exhibits such behavior at a pull position of approximately 20 mm, the boules grown with this technique often exhibit inward growth early in the shoulder.

An alternative approach that avoids the controller cut-back following a flash in boule growth is to grow the initial portion of the shoulder by decreasing the temperature with an open-loop schedule. A temperature schedule based on pull position l_c is implemented to decrease the temperature until the shoulder has grown to approximately one third to one half the final diameter. The closed-loop control is then engaged smoothly using a "bumpless transfer" mechanism.

4.3. Model-Based Compensator Design

The lumped-parameter model developed in Section 3 was used to design a closed-loop compensation algorithm that provides better shape control than the Cambridge Instruments algorithm. The advanced compensation algorithm implements a PID control to give improved performance, stability and reliability. In addition, because the design is based on a model that apparently describes the system dynamics of the LEC GaAs process well, its development required very little experimentation.

The compensator design is based on the linearized dynamics of the process at approximately 130 mm of crystal length. The qualitative objectives for the design of the compensator were taken to be:

1) Zero steady state error in the radius response to radius set-point changes;
2) Improved damping of the natural modes of the process;
3) Enhanced stability margins;
4) Accurate tracking of the desired radius at the shoulder turn.

Objectives 1 and 4 can be achieved by including integral action in the compensator, and by having sufficient low frequency gain. Linear simulation of the controller with a smoothed step command (to emulate the actual desired radius) indicated that the velocity constant (i.e., the intersection of the integrator asymptote with the frequency axis at 0 db gain) should be at least 5×10^{-4} sec^{-1}. Objective 3 can also be achieved by having sufficient gain at the natural frequency of the oscillatory modes. The principal difficulty in the compensator design is achieving the required gain while still providing sufficient phase and gain margins for the closed-loop system. The closed-loop bandwidth will be limited by the phase lag contribution of the nonminimum phase zero at 1.7×10^{-3}. The design must trade-off between increased low-frequency gain and additional phase lag due to the nonminimum phase zero.

The compensator design accomplishes this tradeoff with lead compensation provided by a lightly damped zero pair at the approximate frequency of the dominant plant pole pair (natural frequency 0.0007 rad/sec, damping ratio 0.4) together with a first order lag at the same frequency. The compensator transfer function is:

$$K_1(s) = 0.03\left(\frac{s^2+2(0.4)\left(7\times10^{-4}\right)s+\left(7\times10^{-4}\right)^2}{s\left(s+7\times10^{-4}\right)}\right)K_{dT}K_w \qquad (10)$$

where K_{dT} is the scale constant that converts the drive signal units (volts) to the model input units (°K) and K_w is the scale factor that converts the model weight measurement units (kg) into the weight cell measurement units (volts):

$$K_{dT} = 187 \text{ °K/volt} \qquad K_w = 47047 \text{ volt/kg}$$

Thus, the compensator that is implemented on the process is $K_1(s)$ without the scale factors.

Bode plots of the compensated loop transfer function are shown in Fig. 4. The gain crossover is 0.0012 rad/sec with a phase margin of 25°. The phase crossover frequency is 0.0029 rad/sec with a gain margin of 1.7.

This compensator was used to grow boule AGA8422. The compensator transfer function was discretized using a bilinear transformation with a sampling period of 10 sec. Because the sampling frequency (0.63 rad) is more than 100 times the closed-loop bandwidth, the effects of sampling are negligible. Also, analysis of the linearized model throughout the body growth indicates that the process gain can be expected to increase as the length of the boule increases. Thus, a gain schedule was implemented to reduce the controller gain when the radius stabilizes following the shoulder growth (at 110 mm of boule length).

This control algorithm used an open-loop temperature command schedule to grow the initial shoulder, and the closed-loop compensation described in this section. The radius profile of the boule is shown in Fig. 5, and a photograph of the boule is shown in Fig. 6.

Overall, the shape of boule AGA8422 is superior to that of the boules grown by the analog control system. The radius is controlled throughout the body to within ±0.75 mm variation from the nominal value. The radius during shoulder growth uniformly increases, and transitions smoothly to the nominal body value with a minimal overshoot.

Besides the improved boule shape, the control algorithm allowed the process to tolerate a significant equipment malfunction during the growth. At approximately 40 mm of crystal length, the susceptor began to rub against the guide ring. In a successful effort to eliminate the vibration, the crucible rotation rate was reduced to zero, and then increased slowly until the nominal rotation rate was regained. Normally, such action would cause the boule to pull free of the melt. However, the control system was able to maintain the crystal growth with only a minor perturbation in radius.

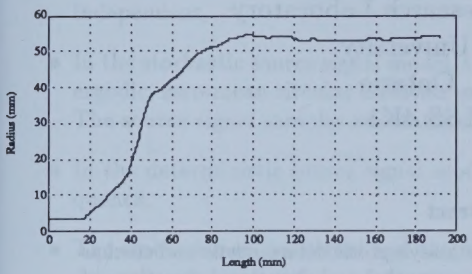


Figure 5. Radius vs. length for boule AGA8422

Figure 4.10. Photograph of boule AGA8422.

References

[1] W. Bardsley, B. Cockayne, G. W. Green, D. T. J. Hurle, G. J. Joyce, J. M. Roslington, P. T. Tufton, H. C. Webber, and M. Healey, *J. Crystal Growth* **24/25** (1974) 369.

[2] W. Bardsley, D. T. J. Hurle and G. C. Joyce, *J. Crystal Growth* **40** (1977) 13.

[3] W. Bardsley, D. T. J. Hurle, G. C. Joyce, and G. C. Wilson, *J. Crystal Growth* **40** (1977) 21.

[4] D. T. J. Hurle, *J. Crystal Growth* **42** (1977) 473.

[5] T. H. Johansen, *J. Crystal Growth* **80** (1987) 343.

[6] T. H. Johansen, *J. Crystal Growth* **84** (1987) 609.

[7] M. A. Gevelber and G. Stephanopoulos, *J. Crystal Growth* **84** (1987) 647.

[8] M. A. Gevelber and G. Stephanopoulos, *J. Crystal Growth* **85** (1987) 256.

[9] D. T. J. Hurle, G. C. Joyce, M. Ghassempoory, A. B. Crowley, and E. J. Stern, J. *Crystal Growth* **100** (1990) 11.

[10] G. A. Satunkin and A. G. Leonov, *J. Crystal Growth* **102** (1990) 592.

[11] A. S. Jordan and Parsley, J. M., Jr., *MRS Bulletin*, **XIII**, (1988), 36.

[12] J. J. Derby, *MRS Bulletin*, **XIII**, (1988), 29.

[13] D. P. Looze, F. Azad, B. Bernstein, and T. Collins, *JCG*, (to appear).

[14] J. S. Freudenberg and D. P. Looze, *IEEE Trans. A. C.*, **AC-30** (1985) 555.

Cramér-Rao Bounds for Two-Dimensional DOA Estimation Using Uniform Circular Arrays

Albert Y.J. Chan and John Litva
Communications Research Laboratory
McMaster University
Hamilton, Ontario
Canada L8S 4K1

Abstract

We derive analytical expressions for both asymptotic deterministic and stochastic Cramér-Rao lower bounds in the joint azimuthal and elevational directions-of-arrival estimation using uniform circular arrays. As well, we show Cramér-Rao lower bound properties with numerical examples.

1 Introduction

The accuracy of direction-of-arrival (DOA) estimate is usually measured by the Cramér-Rao lower bound (CRLB). In this paper, we extend the results in Stoica [1, 2] by deriving asymptotically the stochastic and the deterministic CRLB's, which are based on the stochastic and the deterministic source signal models respectively, for the joint azimuthal and elevational directions-of-arrival (AEDOA) estimation using uniform circular arrays.

The uniform circular array consists of M identical omnidirectional sensors evenly spaced on a ring of radius R on the X-Y plane with its centre at the origin of the Cartesian coordinate system, as shown in Figure 1(a). The position vector of m th sensor is $\mathbf{r}_m = (R\cos\phi_m,\ R\sin\phi_m,\ 0)$ where $\phi_m = 2\pi(m-1)/M$ is the angular direction measured counter-clockwise from the X-axis. Assume that there are K narrow-band plane waves with carrier wavelength λ_c impinging on the array. The azimuth angle α_k and elevation angle β_k of k th radiating source are illustrated in Figure 1(b). Using the complex envelope representation, the observed signal vector $\mathbf{x}(t)$ in the presence of additive narrow-band noise $\mathbf{n}(t)$ is given by

$$\mathbf{x}(t) = \mathbf{S}\mathbf{a}(t) + \mathbf{n}(t) \tag{1}$$

where $\mathbf{a}(t) = [a_1(t)\ a_2(t)\ldots\ a_K(t)]^T$ is the source signal vector, $\mathbf{S} = [\mathbf{s}_1\ \mathbf{s}_2\ldots\ \mathbf{s}_K]$ is the M-by-K steering matrix with $\mathbf{s}_k$ given by

$$\mathbf{s}_k = \begin{bmatrix} \exp(\frac{j2\pi R}{\lambda_c}\sin\beta_k\cos(\alpha_k-\phi_1)) \\ \exp(\frac{j2\pi R}{\lambda_c}\sin\beta_k\cos(\alpha_k-\phi_2)) \\ \vdots \\ \exp(\frac{j2\pi R}{\lambda_c}\sin\beta_k\cos(\alpha_k-\phi_M)) \end{bmatrix}, \qquad k = 1,\ 2,\ldots,\ K \tag{2}$$

and the superscript T denotes the transpose. In practical situations, the ensemble-averaged covariance matrix $\mathbf{R}_x = E[\mathbf{x}(t)\mathbf{x}^\dagger(t)]$ is not available. Note that the notation $E[\cdot]$ represents the expectation operator. Its estimate under N data snapshots will be $\hat{\mathbf{R}}_x = 1/N \sum_{i=1}^{N} \mathbf{x}(i)\mathbf{x}^\dagger(i)$ where the superscript $\dagger$ denotes the conjugate transpose. Some assumptions made about the data model are in the followings.

- The additive noise is modeled as a stationary, second-order ergodic, zero-mean circular Gaussian process with the covariance matrix $\sigma_n^2\mathbf{I}$ where σ_n^2 is an unknown scalar and $\mathbf{I}$ is an identity matrix. In addition, the noise samples are statistically independent.

- In the stochastic source signal model, the source signal is a stationary, second-order ergodic, zero-mean circular Gaussian process with the source covariance matrix $\mathbf{R}_a$. The source signal samples are statistically independent.

- In the deterministic source signal model, $\{\mathbf{a}(i)\}_{i=1}^{N}$ is a bounded deterministic sequence.

- The stochastic source signal and the noise are uncorrelated.

2 Deterministic CRLB

Under the deterministic source signal model, the set of unknown non-random parameters of interest is $\mathcal{C}_d = \{\sigma_n^2,\ \{\Re\{\mathbf{a}(p)\}\}_{p=1}^{N},\ \{\Im\{\mathbf{a}(p)\}\}_{p=1}^{N},\ \Theta\}$ where

$$\Theta = [\alpha_1\ \alpha_2\ \cdots\ \alpha_K \mid \beta_1\ \beta_2\ \cdots\ \beta_K]^T = [\boldsymbol{\alpha} \mid \boldsymbol{\beta}]^T \tag{3}$$

Note that $\Re\{\cdot\}$ and $\Im\{\cdot\}$ denote the real part and the imaginary part of the operand, respectively. Let us define $\mathbf{S}_\alpha$, $\mathbf{S}_\beta$ and the diagonal matrix $\mathbf{A}_d(i)$ as

$$\mathbf{S}_\alpha = \left[\frac{\partial \mathbf{s}_1}{\partial \alpha_1}\ \frac{\partial \mathbf{s}_2}{\partial \alpha_2}\ \cdots\ \frac{\partial \mathbf{s}_K}{\partial \alpha_K}\right] \tag{4}$$

$$\mathbf{S}_\beta = \left[\frac{\partial \mathbf{s}_1}{\partial \beta_1}\ \frac{\partial \mathbf{s}_2}{\partial \beta_2}\ \cdots\ \frac{\partial \mathbf{s}_K}{\partial \beta_K}\right] \tag{5}$$

and

$$\mathbf{A}_d(i) = \mathrm{diag}(a_1(i),\ a_2(i),\ \cdots,\ a_K(i)) \tag{6}$$

The log-likelihood function $\ln L_c$ of N independent sample vectors $\{\mathbf{x}(i)\}_{i=1}^{N}$ can be expressed as

$$\ln L_c = -MN\ln(\pi) - MN\ln(\sigma_n^2) - \frac{1}{\sigma_n^2}\sum_{i=1}^{N}[\mathbf{x}(i) - \mathbf{S}\mathbf{a}(i)]^\dagger[\mathbf{x}(i) - \mathbf{S}\mathbf{a}(i)] \tag{7}$$

The first-order partial derivatives of $\ln L_c$ with respect to parameters in $\mathcal{C}_d$ are represented by

$$\mathbf{z}_{\psi_p} = \frac{\partial(\ln L_c)}{\partial(\Re\{\mathbf{a}(p)\})} = \frac{2}{\sigma_n^2}\Re\{\mathbf{S}^\dagger\mathbf{n}(p)\}, \tag{8}$$

$$\mathbf{z}_{\xi_p} = \frac{\partial(\ln L_c)}{\partial(\Im\{\mathbf{a}(p)\})} = \frac{2}{\sigma_n^2}\Im\{\mathbf{S}^\dagger\mathbf{n}(p)\}, \tag{9}$$

$$\mathbf{z}_\alpha = \frac{\partial(\ln L_c)}{\partial \alpha} = \frac{2}{\sigma_n^2} \sum_{i=1}^{N} \Re\left\{\mathbf{A}_d^\dagger(i)\mathbf{S}_\alpha^\dagger \mathbf{n}(i)\right\}, \tag{10}$$

$$\mathbf{z}_\beta = \frac{\partial(\ln L_c)}{\partial \beta} = \frac{2}{\sigma_n^2} \sum_{i=1}^{N} \Re\left\{\mathbf{A}_d^\dagger(i)\mathbf{S}_\beta^\dagger \mathbf{n}(i)\right\} \tag{11}$$

and

$$\mathbf{z}_\sigma = \frac{\partial(\ln L_c)}{\partial(\sigma_n^2)} = \frac{-MN}{\sigma_n^2} + \frac{1}{(\sigma_n^2)^2} \sum_{i=1}^{N} \mathbf{n}^\dagger(i)\mathbf{n}(i) \tag{12}$$

with $p = 1, 2, \ldots, N$. In the expanded form, the Fisher's information matrix $\mathbf{J} = E[\mathbf{z}\mathbf{z}^T]$, where $\mathbf{z} = [\mathbf{z}_\sigma \ \mathbf{z}_{\psi_1}^T \ \mathbf{z}_{\xi_1}^T \ \cdots \ \mathbf{z}_{\psi_N}^T \ \mathbf{z}_{\xi_N}^T \ \mathbf{z}_\alpha^T \ \mathbf{z}_\beta^T]^T$, can be partitioned into

$$\mathbf{J} = \left[\begin{array}{c|c} \mathbf{C}_1 & \mathbf{C}_2 \\ \hline \mathbf{C}_3 & \mathbf{C}_4 \end{array}\right] = \left[\begin{array}{cccccc|cc} \mathbf{J}_{\sigma\sigma} & \mathbf{J}_{\sigma\psi_1} & \mathbf{J}_{\sigma\xi_1} & \cdots & \mathbf{J}_{\sigma\psi_N} & \mathbf{J}_{\sigma\xi_N} & \mathbf{J}_{\sigma\alpha} & \mathbf{J}_{\sigma\beta} \\ \mathbf{J}_{\psi_1\sigma} & \mathbf{J}_{\psi_1\psi_1} & \mathbf{J}_{\psi_1\xi_1} & \cdots & \mathbf{J}_{\psi_1\psi_N} & \mathbf{J}_{\psi_1\xi_N} & \mathbf{J}_{\psi_1\alpha} & \mathbf{J}_{\psi_1\beta} \\ \mathbf{J}_{\xi_1\sigma} & \mathbf{J}_{\xi_1\psi_1} & \mathbf{J}_{\xi_1\xi_1} & \cdots & \mathbf{J}_{\xi_1\psi_N} & \mathbf{J}_{\xi_1\xi_N} & \mathbf{J}_{\xi_1\alpha} & \mathbf{J}_{\xi_1\beta} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{J}_{\psi_N\sigma} & \mathbf{J}_{\psi_N\psi_1} & \mathbf{J}_{\psi_N\xi_1} & \cdots & \mathbf{J}_{\psi_N\psi_N} & \mathbf{J}_{\psi_N\xi_N} & \mathbf{J}_{\psi_N\alpha} & \mathbf{J}_{\psi_N\beta} \\ \mathbf{J}_{\xi_N\sigma} & \mathbf{J}_{\xi_N\psi_1} & \mathbf{J}_{\xi_N\xi_1} & \cdots & \mathbf{J}_{\xi_N\psi_N} & \mathbf{J}_{\xi_N\xi_N} & \mathbf{J}_{\xi_N\alpha} & \mathbf{J}_{\xi_N\beta} \\ \hline \mathbf{J}_{\alpha\sigma} & \mathbf{J}_{\alpha\psi_1} & \mathbf{J}_{\alpha\xi_1} & \cdots & \mathbf{J}_{\alpha\psi_N} & \mathbf{J}_{\alpha\xi_N} & \mathbf{J}_{\alpha\alpha} & \mathbf{J}_{\alpha\beta} \\ \mathbf{J}_{\beta\sigma} & \mathbf{J}_{\beta\psi_1} & \mathbf{J}_{\beta\xi_1} & \cdots & \mathbf{J}_{\beta\psi_N} & \mathbf{J}_{\beta\xi_N} & \mathbf{J}_{\beta\alpha} & \mathbf{J}_{\beta\beta} \end{array}\right] \tag{13}$$

where $\mathbf{J}_{\spadesuit\clubsuit} = E[\mathbf{z}_\spadesuit \mathbf{z}_\clubsuit^T] = \mathbf{J}_{\clubsuit\spadesuit}^T$ for $\spadesuit, \clubsuit = \sigma, \psi_1, \xi_1, \ldots, \psi_N, \xi_N, \alpha, \beta$. After some mathematical manipulations, we get

$$\mathbf{J}_{\sigma\sigma} = \frac{MN}{(\sigma_n^2)^2}, \tag{14}$$

$$\mathbf{J}_{\sigma\psi_p} = \mathbf{J}_{\sigma\xi_p} = \mathbf{J}_{\sigma\alpha} = \mathbf{J}_{\sigma\beta} = \mathbf{0}, \tag{15}$$

$$\mathbf{J}_{\psi_p\psi_i} = \mathbf{J}_{\xi_p\xi_i} = \frac{2}{\sigma_n^2}\Re\{\mathbf{S}^\dagger\mathbf{S}\}\delta_{ip}, \tag{16}$$

$$\mathbf{J}_{\psi_p\xi_i} = \frac{-2}{\sigma_n^2}\Im\{\mathbf{S}^\dagger\mathbf{S}\}\delta_{ip}, \tag{17}$$

$$\mathbf{J}_{\psi_p\alpha} = \frac{2}{\sigma_n^2}\Re\{\mathbf{S}^\dagger\mathbf{S}_\alpha\mathbf{A}_d(p)\}, \tag{18}$$

$$\mathbf{J}_{\psi_p\beta} = \frac{2}{\sigma_n^2}\Re\{\mathbf{S}^\dagger\mathbf{S}_\beta\mathbf{A}_d(p)\}, \tag{19}$$

$$\mathbf{J}_{\xi_p\alpha} = \frac{2}{\sigma_n^2}\Im\{\mathbf{S}^\dagger\mathbf{S}_\alpha\mathbf{A}_d(p)\}, \tag{20}$$

$$\mathbf{J}_{\xi_p\beta} = \frac{2}{\sigma_n^2}\Im\{\mathbf{S}^\dagger\mathbf{S}_\beta\mathbf{A}_d(p)\}, \tag{21}$$

$$\mathbf{J}_{\alpha\alpha} = \frac{2}{\sigma_n^2} \sum_{i=1}^{N} \Re\left\{\mathbf{A}_d^\dagger(i)\mathbf{S}_\alpha^\dagger\mathbf{S}_\alpha\mathbf{A}_d(i)\right\}, \tag{22}$$

$$\mathbf{J}_{\alpha\beta} = \frac{2}{\sigma_n^2} \sum_{i=1}^{N} \Re\left\{\mathbf{A}_d^\dagger(i)\mathbf{S}_\alpha^\dagger\mathbf{S}_\beta\mathbf{A}_d(i)\right\}, \tag{23}$$

and

$$\mathbf{J}_{\beta\beta} = \frac{2}{\sigma_n^2} \sum_{i=1}^{N} \Re\left\{ \mathbf{A}_d^\dagger(i) \mathbf{S}_\beta^\dagger \mathbf{S}_\beta \mathbf{A}_d(i) \right\} \tag{24}$$

Note that δ_{ip} and $\mathbf{0}$ denote the Kronecker delta and the zero matrix, respectively. Making use of the matrix inversion formula, the asymptotic deterministic CRLB's for estimation of Θ are imbedded in the principle diagonal of $\mathbf{CR}_d$ given by

$$\begin{aligned} \mathbf{CR}_d &= \lim_{N\to\infty} \left[\mathbf{C}_4 - \mathbf{C}_3 \mathbf{C}_1^{-1} \mathbf{C}_2 \right]^{-1} \\ &= \frac{\sigma_n^2}{2N} \left\{ \Re\left\{ \left[\begin{array}{c|c} \mathbf{\Lambda}_1 & \mathbf{\Lambda}_2 \\ \hline \mathbf{\Lambda}_3 & \mathbf{\Lambda}_4 \end{array} \right] \odot \left[\begin{array}{c|c} \mathbf{R}_a & \mathbf{R}_a \\ \hline \mathbf{R}_a & \mathbf{R}_a \end{array} \right]^T \right\} \right\}^{-1} \end{aligned} \tag{25}$$

where

$$\mathbf{\Lambda}_1 = \mathbf{S}_\alpha^\dagger \mathbf{P}_S^\perp \mathbf{S}_\alpha, \tag{26}$$

$$\mathbf{\Lambda}_2 = \mathbf{S}_\alpha^\dagger \mathbf{P}_S^\perp \mathbf{S}_\beta, \tag{27}$$

$$\mathbf{\Lambda}_3 = \mathbf{S}_\beta^\dagger \mathbf{P}_S^\perp \mathbf{S}_\alpha, \tag{28}$$

$$\mathbf{\Lambda}_4 = \mathbf{S}_\beta^\dagger \mathbf{P}_S^\perp \mathbf{S}_\beta, \tag{29}$$

$\odot$ denotes the element-by-element multiplication operator and $\mathbf{P}_S^\perp$ is the projector onto the orthogonal complement of the column space of $\mathbf{S}$ defined as

$$\mathbf{P}_S^\perp = \mathbf{I} - \mathbf{S}(\mathbf{S}^\dagger \mathbf{S})^{-1} \mathbf{S}^\dagger \tag{30}$$

3 Stochastic CRLB

Under the stochastic source signal model, the set of unknown non-random parameters of interest is $\mathcal{C}_s = \{\sigma_n^2,\ \mathbf{R}_a,\ \Theta\}$. Under mild regularity conditions, the stochastic maximum likelihood (ML) estimator achieves the stochastic CRLB asymptotically [3]. In a sense, the asymptotic stochastic CRLB matrix $\mathbf{CR}_s$ is equal to the asymptotic covariance matrix $\mathbf{\Phi}_s$ of the stochastic ML estimator. Based on Böhme [4, 5], the stochastic ML estimate $\breve{\Theta}$ can be obtained by minimizing a criterion function h_s over only Θ. The stochastic ML estimates $\breve{\mathbf{R}}_a$ and $\breve{\sigma}_n^2$ are then calculated by inserting the stochastic ML estimate $\breve{\Theta}$ into explicit formulae. The stochastic ML criterion function h_s and the explicit formulae are given in the followings where $\mathrm{tr}(\cdot)$ and $\det(\cdot)$ denote the trace and the determinant operators, respectively.

$$h_s = \ln\left[\det(\mathbf{S} \breve{\mathbf{R}}_a \mathbf{S}^\dagger + \breve{\sigma}_n^2 \mathbf{I}) \right] \tag{31}$$

$$\breve{\mathbf{R}}_a = (\mathbf{S}^\dagger \mathbf{S})^{-1} \mathbf{S}^\dagger \hat{\mathbf{R}}_x \mathbf{S} (\mathbf{S}^\dagger \mathbf{S})^{-1} - \breve{\sigma}_n^2 (\mathbf{S}^\dagger \mathbf{S})^{-1} \tag{32}$$

$$\breve{\sigma}_n^2 = \frac{1}{M-K} \mathrm{tr}(\mathbf{P}_S^\perp \hat{\mathbf{R}}_x) \tag{33}$$

Based on [6], we see that the stochastic ML estimates $\{\breve{\sigma}_n^2,\ \breve{\mathbf{R}}_a,\ \breve{\Theta}\}$ converges with probability one to the true values as $N \to \infty$. Let ∇_Θ denote the gradient operator defined as

$$\nabla_\Theta = \left[\frac{\partial}{\partial \alpha_1} \ \cdots \ \frac{\partial}{\partial \alpha_K} \ \frac{\partial}{\partial \beta_1} \ \cdots \ \frac{\partial}{\partial \beta_K} \right]^T \tag{34}$$

Invoking the ML principle, $\breve{\Theta}$ is given by $(\nabla_\Theta h_s)_{\breve{\Theta}} = \mathbf{0}$. Assume that h_s is twice-differentiable. The Taylor expansion of $(\nabla_\Theta h_s)_{\breve{\Theta}}$ around Θ_0 yields

$$\mathbf{0} = (\nabla_\Theta h_s)_{\Theta_0} + \left[\nabla_\Theta (\nabla_\Theta h_s)^T\right]_{\Theta_\xi} (\breve{\Theta} - \Theta_0) \tag{35}$$

where Θ_ξ is a vector of some value on the line segment joining $\breve{\Theta}$ and Θ_0. Define $\mathbf{G}_s$ as

$$\mathbf{G}_s = \lim_{N\to\infty} \left[\nabla_\Theta (\nabla_\Theta h_s)^T\right]_{\Theta_0} \tag{36}$$

which is assumed to be non-singular. Using the multivariate central limit theorem, we see that $(\breve{\Theta} - \Theta_0)$ is asymptotically Gaussian distributed with a zero mean vector. Its covariance matrix $\mathbf{\Phi}_s(\Theta)$ is of the form

$$\mathbf{\Phi}_s(\Theta) = (\mathbf{G}_s)^{-1} \lim_{N\to\infty} E\left[\nabla_\Theta h_s (\nabla_\Theta h_s)^T\right]_{\Theta_0} (\mathbf{G}_s)^{-1} \tag{37}$$

Based on [7], for $1 \le i \le K$ and $\spadesuit \in \{\alpha,\ \beta\}$, we have

$$\frac{\partial h_s}{\partial \spadesuit_i} = \mathrm{tr}\left[\left(\mathbf{S}\breve{\mathbf{R}}_a\mathbf{S}^\dagger + \breve{\sigma}_n^2\mathbf{I}\right)^{-1} \frac{\partial}{\partial \spadesuit_i}\left(\mathbf{S}\breve{\mathbf{R}}_a\mathbf{S}^\dagger + \breve{\sigma}_n^2\mathbf{I}\right)\right] \tag{38}$$

Making use of

$$\mathbf{P}_S = \mathbf{S}\left(\mathbf{S}^\dagger\mathbf{S}\right)^{-1}\mathbf{S}^\dagger, \tag{39}$$

$$\frac{\partial \mathbf{P}_S}{\partial \spadesuit_i} = \mathbf{P}_S\frac{\partial \mathbf{P}_S}{\partial \spadesuit_i} + \left(\mathbf{P}_S\frac{\partial \mathbf{P}_S}{\partial \spadesuit_i}\right)^\dagger, \tag{40}$$

$$\frac{\partial \mathbf{P}_S^\perp}{\partial \spadesuit_i} = \mathbf{P}_S^\perp\frac{\partial \mathbf{P}_S^\perp}{\partial \spadesuit_i} + \left(\mathbf{P}_S^\perp\frac{\partial \mathbf{P}_S^\perp}{\partial \spadesuit_i}\right)^\dagger \tag{41}$$

and

$$\frac{\partial \mathbf{P}_S}{\partial \spadesuit_i}\mathbf{S} = \mathbf{P}_S^\perp\frac{\partial \mathbf{S}}{\partial \spadesuit_i}, \tag{42}$$

we can determine

$$\frac{\partial h_s}{\partial \spadesuit_i} = 2\Re\left\{\mathrm{tr}\left\{\left[\left(\mathbf{S}^\dagger\hat{\mathbf{R}}_x\mathbf{S}\right)^{-1} - \frac{\left(\mathbf{S}^\dagger\mathbf{S}\right)^{-1}}{\sigma_n^2}\right]\mathbf{S}^\dagger\hat{\mathbf{R}}_x\mathbf{P}_S^\perp\frac{\partial \mathbf{S}}{\partial \spadesuit_i}\right\}\right\} \tag{43}$$

By virtue of

$$\frac{\partial \mathbf{P}_S^\perp}{\partial \spadesuit_i}\mathbf{P}_S^\perp = -\mathbf{S}\left(\mathbf{S}^\dagger\mathbf{S}\right)^{-1}\frac{\partial \mathbf{S}^\dagger}{\partial \spadesuit_i}\mathbf{P}_S^\perp, \tag{44}$$

We have

$$\lim_{N\to\infty} \nabla_\Theta (\nabla_\Theta h_s)^T = \frac{2}{\sigma_n^2}\Re\left\{\left[\begin{array}{c|c} \mathbf{\Lambda}_1 & \mathbf{\Lambda}_2 \\ \hline \mathbf{\Lambda}_3 & \mathbf{\Lambda}_4 \end{array}\right] \odot \left[\begin{array}{c|c} \mathbf{\Xi} & \mathbf{\Xi} \\ \hline \mathbf{\Xi} & \mathbf{\Xi} \end{array}\right]^T\right\} \tag{45}$$

where

$$\mathbf{\Xi} = \mathbf{R}_a\mathbf{S}^\dagger\mathbf{R}_x^{-1}\mathbf{S}\mathbf{R}_a \tag{46}$$

Let the vector $\boldsymbol{\tau}_i$ denote the i th column of the matrix $[\left(\mathbf{S}^\dagger\mathbf{R}_x\mathbf{S}\right)^{-1} - \left(\mathbf{S}^\dagger\mathbf{S}\right)^{-1}/\sigma_n^2]\mathbf{S}^\dagger$. On the basis of [8], for $1 \le i,k \le K$ and $\spadesuit,\ \clubsuit \in \{\alpha,\ \beta\}$, we have

$$E\left[\boldsymbol{\tau}_k^T\hat{\mathbf{R}}_x\mathbf{P}_S^\perp\frac{\partial \mathbf{s}_k}{\partial \spadesuit_k}\boldsymbol{\tau}_i^T\hat{\mathbf{R}}_x\mathbf{P}_S^\perp\frac{\partial \mathbf{s}_i}{\partial \clubsuit_i}\right] = 0 \tag{47}$$

and

$$E\left[\boldsymbol{\tau}_k^T \hat{\mathbf{R}}_x \mathbf{P}_S^{\perp} \frac{\partial \mathbf{s}_k}{\partial \spadesuit_k} \frac{\partial \mathbf{s}_i^{\dagger}}{\partial \clubsuit_i} \mathbf{P}_S^{\perp} \hat{\mathbf{R}}_x \boldsymbol{\tau}_i^*\right] = \frac{1}{N\sigma_n^2} [\boldsymbol{\Xi}]_{ki} \left[\mathbf{S}_{\clubsuit}^{\dagger} \mathbf{P}_S^{\perp} \mathbf{S}_{\spadesuit}\right]_{ik} \tag{48}$$

where the notation $[\cdot]_{ik}$ and the superscript $*$ denote the (i,k) element of the operand matrix and the conjugate, respectively. Accordingly, we obtain

$$\lim_{N\to\infty} E\left[(\nabla_\Theta h_s)(\nabla_\Theta h_s)^T\right] = \frac{2}{N\sigma_n^2} \Re\left\{\left[\begin{array}{c|c} \boldsymbol{\Lambda}_1 & \boldsymbol{\Lambda}_2 \\ \hline \boldsymbol{\Lambda}_3 & \boldsymbol{\Lambda}_4 \end{array}\right] \odot \left[\begin{array}{c|c} \boldsymbol{\Xi} & \boldsymbol{\Xi} \\ \hline \boldsymbol{\Xi} & \boldsymbol{\Xi} \end{array}\right]^T\right\} \tag{49}$$

Thus, $\mathbf{CR}_s$ can be expressed as

$$\mathbf{CR}_s = \boldsymbol{\Phi}_s(\Theta) = \frac{\sigma_n^2}{2N} \left\{\Re\left\{\left[\begin{array}{c|c} \boldsymbol{\Lambda}_1 & \boldsymbol{\Lambda}_2 \\ \hline \boldsymbol{\Lambda}_3 & \boldsymbol{\Lambda}_4 \end{array}\right] \odot \left[\begin{array}{c|c} \boldsymbol{\Xi} & \boldsymbol{\Xi} \\ \hline \boldsymbol{\Xi} & \boldsymbol{\Xi} \end{array}\right]^T\right\}\right\}^{-1} \tag{50}$$

4 Numerical Results

We perform numerical examples to gain some insights into CRLB properties. We consider two equipowered narrow-band plane waves impinging on a 16-element uniform circular array with the inter-sensor spacing equal to $0.5\lambda_c$. The number of data snapshots is 100. Based on the closed form CRLB matrix expressions, we plot the CRLB's for the AEDOA estimates of the first source against

- (a) the signal-to-noise ratio (SNR) under: $(\alpha, \beta) \in \{(20°, 30°), (22°, 32°)\}$ and correlation coefficient = 0.99,
- (b) the correlation coefficient under: $(\alpha, \beta) \in \{(20°, 30°), (22°, 32°)\}$ and SNR = 0 dB,
- (c) the azimuthal DOA separation under: SNR = 0 dB, correlation ceofficient = 0.99, $(\alpha_1, \beta_1) = (20°, 30°)$ and $\beta_2 = 32°$,
- (d) the elevational DOA separation under: SNR = 0 dB, correlation ceofficient = 0.99, $(\alpha_1, \beta_1) = (20°, 30°)$ and $\alpha_2 = 22°$,

as shown in Figures 2 and 3. We note that the deterministic CRLB is always smaller or equal to the stochastic CRLB. In addition, we see that the CRLB's decrease with increasing SNR, but increase with increasing correlation coefficient.

In Figures 2(c) and 3(c), we observe that the CRLB's for the azimuthal DOA estimate are increasing functions when the azimuthal angular separation is less than 2.5°, and decreasing functions otherwise. On the other hand, the CRLB's for the elevational DOA estimate decrease with increasing azimuthal angular separation.

We proceed to the CRLB's for the case (d), as shown in Figures 2(d) and 3(d). The CRLB's for the azimuthal DOA estimate decrease with increasing elevational angular separation. On the other hand, the CRLB's for the elevational DOA estimate are increasing functions when the elevational angular separation is less than 1°, and decreasing functions otherwise.

In conclusion, we established closed-form analytical expressions for asymptotic deterministic and stochastic Cramér-Rao lower bounds for joint AEDOA estimates. They can serve as benchmarks in the performance comparison of AEDOA estimators. In addition, we observed some CRLB properties from numerical examples.

References

[1] P. Stoica and A. Nehorai, "MUSIC, maximum likelihood, and Cramér-Rao bound," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-37, no. 5, pp. 720–741, May 1989.

[2] P. Stoica and A. Nehorai, "Performance study of conditional and unconditional direction-of-arrival estimation," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-38, no. 10, pp. 1783–1795, October 1990.

[3] W.J. Bang, "Array processing with generalized beamformers," Ph.D. dissertation, Yale University, New Haven, Connecticut, 1971.

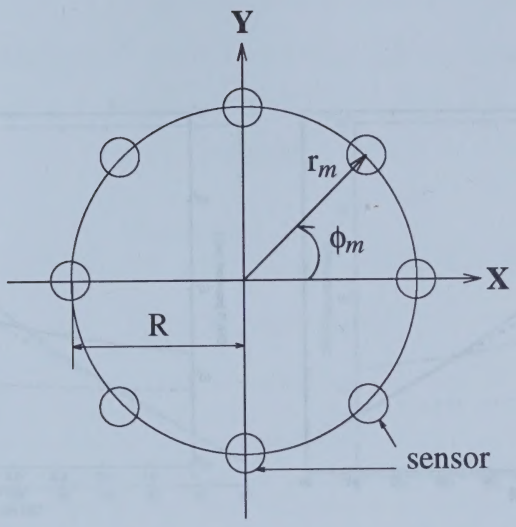
[4] J.F. Böhme, "Estimation of spectral parameters of correlated signals in wavefields," *Signal Processing*, vol. 11, no. 4, pp. 329–337, 1986.

[5] U. Sandkühler and J.F. Böhme, "Accuracy of maximum likelihood estimates for array processing," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Dallas, Texas, 1987, pp. 2015–2018.

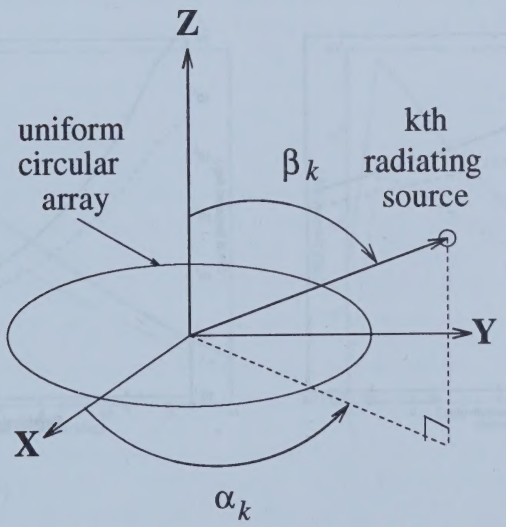
[6] T.W. Anderson, "Multivariate linear relations," in *Proceedings of Second International Tampere Conference on Statistics*, Tampere, Finland, 1987, pp. 9–36.

[7] A. Graham, *Kronecker Products and Matrix Calculus with Applications.* Chichester, England: Ellis Horwood Ltd., 1981.

[8] P. Janssen and P. Stoica, "On the expectation of the product of four matrix-valued Gaussian random variables," *IEEE Transactions on Automatic Control*, vol. AC-33, no. 9, pp. 867–870, September 1988.

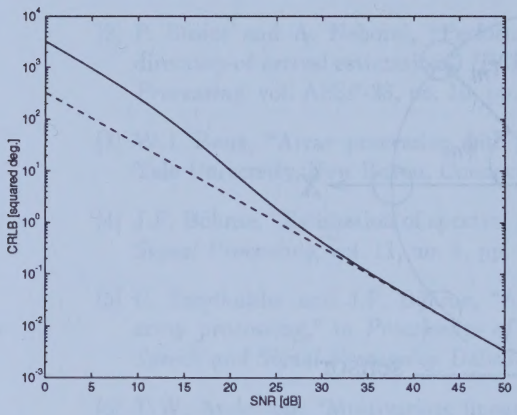


(a)

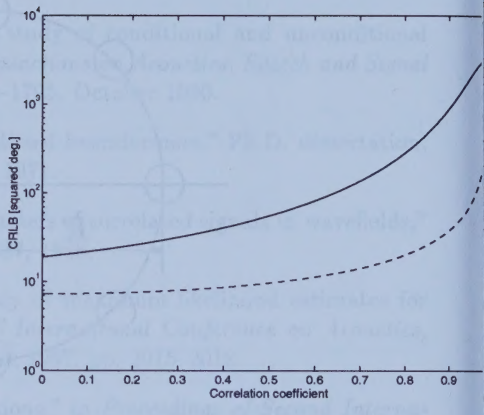


(b)

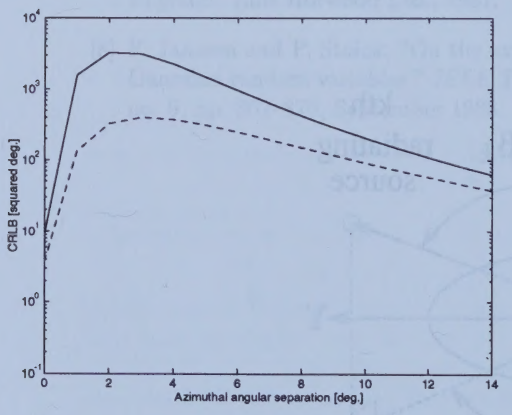
Figure 1: An uniform circular array with the k th radiating source. (a) An uniform circular array of sensors ($M = 8$) lies on the X-Y plane. (b) Azimuth (α_k) and elevation (β_k) angles of the k th radiating source.



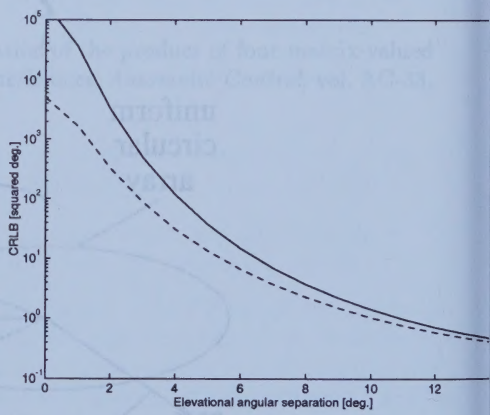
(a)



(b)

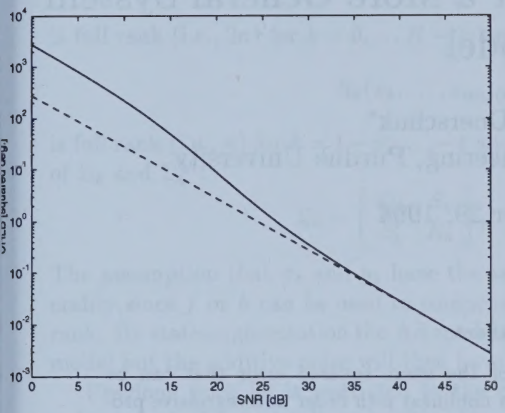


(c)

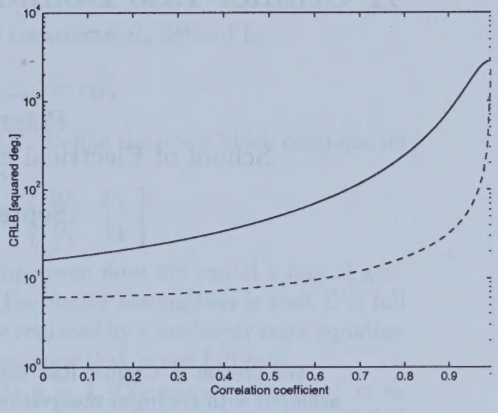


(d)

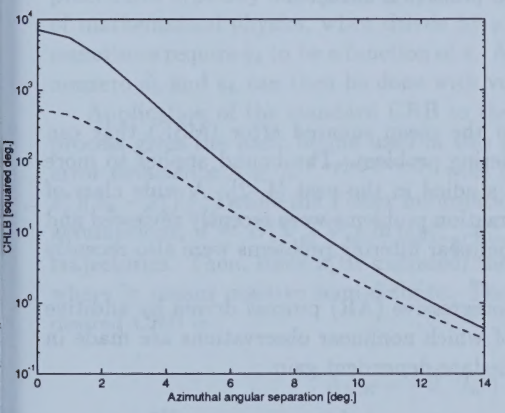
Figure 2: Stochastic (solid line) and deterministic (dashed line) CRLB's for the azimuthal DOA estimate of the first source are plotted versus (a) SNR, (b) correlation coefficient, (c) azimuthal angular separation, and (d) elevational angular separation.



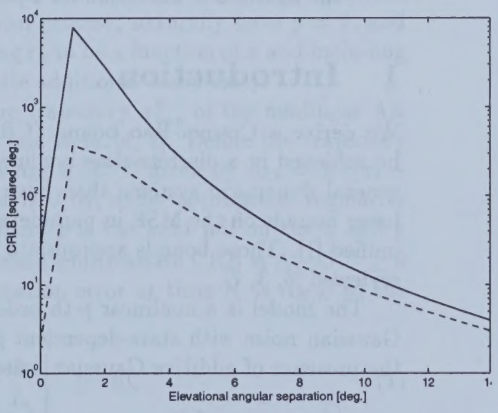
(a)



(b)



(c)



(d)

Figure 3: Stochastic (solid line) and deterministic (dashed line) CRLB's for the elevational DOA estimate of the first source are plotted versus (a) SNR, (b) correlation coefficient, (c) azimuthal angular separation, and (d) elevational angular separation.

Performance of Discrete-Time Nonlinear Filters: A Cramer-Rao Bound for a More General System Model

Peter C. Doerschuk*
School of Electrical Engineering, Purdue University

September 29, 1994

Abstract

We present a Cramer-Rao bound for the mean squared error that can be achieved with nonlinear observations of a nonlinear p-th order autoregressive process where both the process and observation noise covariances can be state dependent. A requirement that the additive white Gaussian process noise have a full-rank covariance is the major limitation. A numerical example demonstrating the tightness of the bound for a particular problem is included.

1 Introduction

We derive a Cramer-Rao bound (CRB) on the mean squared error (MSE) that can be achieved in a discrete-time nonlinear filtering problem. The bound applies to more general dynamical systems than have been studied in the past [4, 2]. A wide class of lower bounds on the MSE in parameter estimation problems were recently reviewed and unified [1]. Those bounds appropriate for nonlinear filtering problems were also recently reviewed [6, 3, 7].

The model is a nonlinear p-th order autoregressive (AR) process driven by additive Gaussian noise with state-dependent gain of which nonlinear observations are made in the presence of additive Gaussian noise with state-dependent gain:

$$\begin{aligned} x_{k+1} &= f_k(x_k, \ldots, x_{k-p+1}) + [q_k(x_k, \ldots, x_{k-p+1}), d_k(x_k, \ldots, x_{k-p+1})]\xi_k, \\ &\qquad \text{for } k = 0, \ldots, K-1 \\ y_k &= h_k(x_k, \ldots, x_{k-p+1}) + [e_k(x_k, \ldots, x_{k-p+1}), r_k(x_k, \ldots, x_{k-p+1})]\xi_k, \\ &\qquad \text{for } k = 0, \ldots, K-1 \\ y_k &= h_k(x_k, \ldots, x_{\max(k-p+1,1-p)}) + r_k(x_k, \ldots, x_{\max(k-p+1,1-p)})v_k, \\ &\qquad \text{for } k = 1-p, \ldots, -1 \text{ and } k = K \end{aligned}$$

*Supported by U. S. National Science Foundation grant MIP-9110919, a Whirlpool Faculty Fellowship, and the School of Electrical Engineering, Purdue University. Mailing address: Peter C. Doerschuk, 1285 Electrical Engineering Building, Purdue University, West Lafayette, IN 47907–1285; Telephone: (317) 494–1742; Fax: (317) 494–6440; Internet: doerschu@ecn.purdue.edu.

where $x_k, y_k \in \mathcal{R}^n$; the range of f_k and h_k is $\mathcal{R}^n$; the range of q_k, d_k, r_k, and e_k is $\mathcal{R}^{n\times n}$; ξ_k is i.i.d. $N(0, I_{2n})$; v_k is i.i.d. $N(0, I_n)$; $(x'_0, \ldots, x'_{1-p})'$ is $p_0(x_0, \ldots, x_{1-p})$ which is never zero; ξ, v and $(x'_0, \ldots, x'_{1-p})'$ are independent; the covariance Σ_k defined by

$$\Sigma_k(x_k, \ldots, x_{k-p+1}) = \begin{bmatrix} q_k & d_k \\ e_k & r_k \end{bmatrix} \begin{bmatrix} q_k & d_k \\ e_k & r_k \end{bmatrix}'$$

is full rank (i.e., $2n$) for $k = 0, \ldots K-1$; and the covariance R_k defined by

$$R_k(x_k, \ldots, x_{\max(k-p+1,1-p)}) = r_k r'_k$$

is full rank (i.e., n) for $k = 1-p, \ldots, -1$ and $k = K$. Define the $n \times n$ block components of Σ_k and Σ_k^{-1}:

$$\Sigma_k = \begin{bmatrix} Q_k & S_k \\ S'_k & R_k \end{bmatrix}, \quad \Sigma_k^{-1} = \begin{bmatrix} W_k & U_k \\ U'_k & V_k \end{bmatrix}.$$

The assumption that x_k and y_k have the same dimension does not entail a loss of generality since f or h can be used to compensate. The major assumption is that Σ is full rank. By state augmentation the AR model can be replaced by a nonlinear state equation model but the additive noise will then have a covariance that is not full rank.

Previous work [4] is restricted to the case (1) $p = 1$, (2) $q_k(x_k, \ldots, x_{k-p+1}) = q_k$ independent of x and full rank, (3) $r_k(x_k, \ldots, x_{k-p+1}) = r_k$ independent of x and full rank, and (4) $d_k(x_k, \ldots, x_{k-p+1}) = e_k(x_k, \ldots, x_{k-p+1}) = 0$. The generalization described here is worthwhile for two reasons: (1) State equations derived from system identification procedures typically have $p > 1$. (2) Discretization of second-order differential equations of mathematical physics, when driven by a random process, naturally have $p = 2$, and sometimes require q_k to be a function of x. Allowing r_k to be a function of x and including nonzero d_k and e_k can then be done with very little additional complexity.

Application of the standard CRB to the entire trajectory x_{1-p}^K of the nonlinear AR process gives the basic bound used in this paper and Refs. [4, 2]. Define the trajectory error covariance $\Lambda \in \mathcal{R}^{(K+p)n\times(K+p)n}$ with blocks $\Lambda_{l,k} \in \mathcal{R}^{n\times n}$ given by $\Lambda_{l,k} = E\{(x_l - \hat{x}_l^*)(x_k - \hat{x}_k^*)'\}$. Define the Fisher information matrix J by, under appropriate regularity assumptions, $J = E\{\nabla_{x_1^K} \nabla_{x_1^K} \ln p(y_{1-p}^K, x_{1-p}^K)\}$ where p is the joint pdf on the x and y trajectories. Then, since $\hat{x}_k^*$ is unbiased, the standard multivariate CRB is $\Lambda - J^{-1} \geq 0$ where $\geq$ means positive semi-definite. The estimation error at time K is $\Lambda_{K,K}$ so the desired CRB is

$$\Lambda_{K,K} - \begin{bmatrix} 0 & I_n \end{bmatrix} J^{-1} \begin{bmatrix} 0 \\ I_n \end{bmatrix} \geq 0. \tag{1}$$

There are two difficulties in applying Eq. 1. The first difficulty is that for a long trajectory the matrix J is large and it is difficult to compute J^{-1}. This problem is circumvented in this paper and Refs. [4, 2] by finding a linear Gaussian system that has the same Fisher information matrix as the nonlinear system of interest. The linear Gaussian system used in this paper will be a p-th order AR process and an observation equation where the observation at time k depends on the AR process at times k, $k-1$, ..., $k-p+1$. This system can be transformed by state augmentation into a linear Gaussian state-variable system. In the state-variable system $\Lambda_{K,K}^{\text{state}}$ can be computed exactly and without excessive computation by the Kalman filter and furthermore the CRB is satisfied with equality so the known value of $\Lambda_{K,K}^{\text{state}}$ is the desired bound on the performance of

any estimator for the nonlinear system. The second difficulty is that the computation of J typically requires numerical computation of expectations. While the approach of this paper and Refs. [4, 2] still requires such computations, the approach organizes the computations so that they can be done by simulation of the nonlinear AR process alone (i.e., not also the observation equation).

The linear Gaussian system used in this paper is a special case of the nonlinear system with

$$\begin{aligned}
f_k(x_{k-p+1}^k) &= \sum_{i=0}^{p-1} A_{k,i} x_{k-i} \\
h_k(x_{\max(k-p+1,1-p)}^k) &= \sum_{i=0}^{\min(p-1,k+p-1)} C_{k,i} x_{k-i} \\
p_0(x_{1-p}^0) &= \mathcal{N}(0, P_0)(x_{1-p}^0) \\
\Sigma_k(x_{k-p+1}^k) &= \check{\Sigma}_k \quad \text{for } k = 0, \ldots, K-1 \text{ independent of } x \\
R_k(x_{\max(k-p+1,1-p)}^k) &= \check{R}_k \quad \text{for } k = 1-p, \ldots, -1 \text{ and } k = K \text{ independent of } x
\end{aligned}$$

where P_0 (P_0^{-1}) has $n \times n$ blocks $P_{i,j}$ ($P^{i,j}$) for i and j in $0, \ldots, 1-p$ and

$$\check{\Sigma}_k = \begin{bmatrix} \check{Q}_k & \check{S}_k \\ \check{S}_k' & \check{R}_k \end{bmatrix}, \qquad \check{\Sigma}_k^{-1} = \begin{bmatrix} \check{W}_k & \check{U}_k \\ \check{U}_k' & \check{V}_k \end{bmatrix}.$$

For $k = 0, \ldots, K-1$ and $l = 0, \ldots, p-1$ define

$$J_{k,l} = \begin{bmatrix} A_{k,l} \\ C_{k,l} \end{bmatrix}.$$

For later convenience, define

$$\begin{aligned}
\Delta_{i,k,l}^{so} &= E\left\{ [\nabla_{x_l} f_i', \nabla_{x_l} h_i'] \, \Sigma_i^{-1} \begin{bmatrix} \nabla_{x_k} f_i \\ \nabla_{x_k} h_i \end{bmatrix} \right\} \\
&\quad + \frac{1}{2} E\left\{ \operatorname{tr}\left[\Sigma_i \frac{\partial^2 \Sigma_i^{-1}}{\partial x_{l,\beta} \partial x_{k,\alpha}} \right] + \nabla_{x_l} \nabla_{x_k} \ln \det \Sigma_i \right\} \qquad (2) \\
\Delta_{i,k,l}^{o} &= E\left\{ \nabla_{x_l} h_i' R_i^{-1} \nabla_{x_k} h_i \right\} + \frac{1}{2} E\left\{ \operatorname{tr}\left[R_i \frac{\partial^2 R_i^{-1}}{\partial x_{l,\beta} \partial x_{k,\alpha}} \right] + \nabla_{x_l} \nabla_{x_k} \ln \det R_i \right\} \qquad (3) \\
\Lambda_{k,l} &= -E\left\{ [\nabla_{x_l} f_k', \nabla_{x_l} h_k'] \Sigma_k^{-1} \begin{bmatrix} I_n \\ 0_n \end{bmatrix} \right\} = -E\left\{ \nabla_{x_l} f_k' W_k + \nabla_{x_l} h_k' U_k' \right\} \qquad (4) \\
\Gamma_k &= E\left\{ [I_n, 0_n] \Sigma_k^{-1} \begin{bmatrix} I_n \\ 0_n \end{bmatrix} \right\} = E\left\{ W_k \right\}. \qquad (5)
\end{aligned}$$

These formulae simplify when, for example, $\Sigma_k(x_{k-p+1}^k)$ is independent of x. In order to determine the linear Gaussian system the user must give values for $\Delta_{i,k,l}^{so}$, $\Delta_{i,k,l}^{o}$, $\Lambda_{k,l}$, Γ_k, and $E\{\nabla_{x_l} \nabla_{x_k} \ln p_0\}$ in the nonlinear system. Computation of the first four will likely require Monte Carlo simulation unless f is linear and p_0 is Gaussian or, for Γ_k, the gain for the process noise is state independent. The exact range of i, k, and l required is determined by Eq. 13.

By equating $E\{\nabla_{x_l} \nabla_{x_k} \ln p\}$ for the nonlinear and the linear-Gaussian systems, where $\ln p$ is the natural logarithm of the joint pdf for the x and y trajectories, we derive a system

of equations that A, C, P_0, $\check{\Sigma}$, and $\check{R}$ must satisfy. It is only necessary to consider $l \leq k$ because $\nabla_{x_l}\nabla_{x_k} \ln p = \nabla_{x_k}\nabla_{x_l} \ln p'$. The equations are

$$
\begin{aligned}
& k = 2-p, \ldots, K; l = \max(k-p, 1-p), \ldots, k-1: \\
& - \sum_{i=\max(k,0)}^{\min(l+p-1,K-1)} \Delta^{so}_{i,k,l} - \Lambda_{k-1,l}\mathcal{I}(k \geq 1) - \Delta^{\circ}_{K,k,l}\mathcal{I}(K-p+1 \leq l) \\
& - \sum_{i=k}^{-1} \Delta^{\circ}_{i,k,l} + E\{\nabla_{x_l}\nabla_{x_k} \ln p_0\}\mathcal{I}(k \leq 0) \\
= & - \sum_{i=\max(k,0)}^{\min(l+p-1,K-1)} J'_{i,i-l}\check{\Sigma}_i^{-1}J_{i,i-k} + \left(A'_{k-1,k-1-l}\check{W}_{k-1} + C'_{k-1,k-1-l}\check{U}'_{k-1}\right)\mathcal{I}(k \geq 1) \\
& - C'_{K,K-l}\check{R}_K^{-1}C_{K,K-k}\mathcal{I}(K-p+1 \leq l) - \sum_{i=k}^{-1} C'_{i,i-l}\check{R}_i^{-1}C_{i,i-k} - P^{l,k}\mathcal{I}(k \leq 0) \qquad (6) \\
& k = 1-p, \ldots, K: \\
& - \sum_{i=\max(k,0)}^{\min(k+p-1,K-1)} \Delta^{so}_{i,k,k} - \Gamma_{k-1}\mathcal{I}(k \geq 1) - \Delta^{\circ}_{K,k,k}\mathcal{I}(K-p+1 \leq k) \\
& - \sum_{i=k}^{-1} \Delta^{\circ}_{i,k,k} + E\{\nabla_{x_k}\nabla_{x_k} \ln p_0\}\mathcal{I}(k \leq 0) \\
= & - \sum_{i=\max(k,0)}^{\min(k+p-1,K-1)} J'_{i,i-k}\check{\Sigma}_i^{-1}J_{i,i-k} - \check{W}_{k-1}\mathcal{I}(k \geq 1) \\
& - C'_{K,K-k}\check{R}_K^{-1}C_{K,K-k}\mathcal{I}(K-p+1 \leq k) - \sum_{i=k}^{-1} C'_{i,i-k}\check{R}_i^{-1}C_{i,i-k} - P^{k,k}\mathcal{I}(k \leq 0) \qquad (7)
\end{aligned}
$$

where the function $\mathcal{I}(e)$ is 1 if e is true and 0 otherwise.

Define in sequence the following quantities:

$$
\begin{aligned}
P^{l,k} &\doteq -E\{\nabla_{x_l}\nabla_{x_k} \ln p_0\} \quad \text{for } k = 1-p, \ldots, 0;\ l = 1-p, \ldots, k. && (8) \\
\check{W}_k &\doteq \Gamma_k \quad \text{for } k = 0, \ldots, K-1. && (9) \\
A_{k,k-l} &\doteq -\check{W}_k^{-1}\Lambda'_{k,l} \quad \text{for } k = 0, \ldots, K-1;\ l = k-p+1, \ldots, k. && (10) \\
\check{U}_k &\doteq 0 \quad \text{for } k = 0, \ldots, K-1. && (11) \\
\check{R}_k &\doteq \text{arbitrary positive definite matrix} \quad \text{for } k = 1-p, \ldots, K. && (12) \\
D_{k,l} &\doteq \sum_{i=\max(k,0)}^{\min(l+p-1,K-1)} \left(\Delta^{so}_{i,k,l} - A'_{i,i-l}\check{W}_i A_{i,i-k}\right) + \Delta^{\circ}_{K,k,l}\mathcal{I}(K-p+1 \leq l) + \sum_{i=k}^{-1} \Delta^{\circ}_{i,k,l} \\
& \quad \text{for } k = 1-p, \ldots, K;\ l = \max(k-p, 1-p), \ldots, k. && (13)
\end{aligned}
$$

Use these definitions in Eqs. 6 and 7 to get

$$
\begin{aligned}
\sum_{i=k}^{\min(l+p-1,K)} C'_{i,i-l}\check{R}_i^{-1}C_{i,i-k} &= D_{k,l} \text{ for } \begin{array}{l} k = 2-p, \ldots, K; \\ l = \max(k-p, 1-p), \ldots, k-1 \end{array} && (14) \\
\sum_{i=k}^{\min(k+p-1,K)} C'_{i,i-k}\check{R}_i^{-1}C_{i,i-k} &= D_{k,k} \text{ for } k = 1-p, \ldots, K. && (15)
\end{aligned}
$$

In spite of being quadratic in $C_{i,j}$, Eqs. 14 and 15 can be solved recursively for $C_{i,j}$. The procedure is given in Figure 1. Once A, C, P_0, $\check{\Sigma}$, and $\check{R}$ have been determined by Eqs. 8,

$$
\begin{aligned}
&\text{for}(k = K; k >= 1 - p; k - -)\{ \\
&\quad C_{k,0} = \check{R}_k^{1/2} \left(D_{k,k} - \sum_{i=k+1}^{\min(k+p-1,K)} C'_{i,i-k} \check{R}_i^{-1} C_{i,i-k} \right)^{T/2} \\
&\quad \text{for}(l = k - 1; l >= \max(k - p + 1, 1 - p); l - -)\{ \\
&\qquad C_{k,k-l} = \left[\left(D_{k,l} - \sum_{i=k+1}^{\min(l+p-1,K)} C'_{i,i-l} \check{R}_i^{-1} C_{i,i-k} \right) \left(\check{R}_k^{-1} C_{k,0} \right)^{-1} \right]' \\
&\quad \} \\
&\}
\end{aligned}
$$

Figure 1: Algorithm for the Solution of Eqs. 14 and 15. The control structures are written in the C programming language. Quantities of the form $^{1/2}$ are matrix square roots: $R = R^{1/2}\left(R^{1/2}\right)'$, $R^{T/2} \doteq \left(R^{1/2}\right)'$.

9, 10, 11, 12, 13, 14, and 15 using the algorithm described in Figure 1, then the state augmentation and Kalman filter computations needed in order to determine the bound are standard.

Evaluation of these equations for $p = 1$ with $d = e = 0$, both $Q_k(x^k_{k-p+1})$ and $R_k(x^k_{\max(k-p+1,1-p)})$ independent of x, and $\check{R}_k = R_k$ recovers the results of Ref. [4]. (There is a typographical error in [4, Eq. 7f] which reads $P_0 = \nabla_{x_0} \nabla_{x_0} p_{x_0}(x_0)$ but should read $-P_0^{-1} = E\{\nabla_{x_0} \nabla_{x_0} \ln p_{x_0}(x_0)\}$).

2 Example

In this section we consider an example estimation problem in order to demonstrate the computations involved and that the bound is tight at least for some estimation problems. (Please contact the author for copies of the software). The physical system is a damped pendulum driven by a random torque where noisy measurements are made of the horizontal component of the pendulum bob's location and the goal is to estimate the angular location of the pendulum (measured from the negative-going vertical). This system demonstrates $p = 2$ and nonlinear state and observation equations. After discretization the equations are

$$
\begin{aligned}
\phi_{n+1} &= \phi_n \left(2 - \frac{\gamma T}{l^2 m}\right) + \phi_{n-1} \left(\frac{\gamma T}{l^2 m} - 1\right) - \frac{g}{l} T^2 \sin \phi_{n-1} + \sigma_w \frac{T^2}{l^2 m} \frac{\tau((n-1)T)}{\sigma_w} \\
&\qquad \text{for } n = 0, \ldots, K-1 \\
y_n &= l \sin \phi_n + v(nT) \qquad \text{for } n = 0, \ldots, K
\end{aligned}
$$

where l is the length of the pendulum, m is the mass of the bob, γ is the coefficient of friction, g is the acceleration due to gravity, T is the sampling interval, τ is the random applied torque ($\mathcal{N}(0, \sigma_w^2)$), and v is the observation noise ($\mathcal{N}(0, \sigma_v^2)$). The MSE of the optimal filter was underbounded by the CRB developed in this note and overbounded by the performance of the Extended Kalman Filter (EKF). When computing expectations for the CRB and the EKF by Monte Carlo methods, the state trajectories used were identical. Monte Carlo evaluation of the performance of the EKF must, however, also

include realizations of the observation trajectory. We consider three cases and for each case compute the CRB and the MSE for the EKF at each point in a $K = 500$ point trajectory for a variety of σ_v. Notice that only one set of expectations need be computed for the CRB in order to determine the bound for any σ_v. The parameters are $l = 1$, $m = 1$, $\gamma = 1$, $g = 10$, and $T = .01$.

In the first and second cases, the pdf on the initial condition is jointly Gaussian: $p(\phi_{-1}, \phi_0) = \mathcal{N}(0, \sigma_\phi^2)(\phi_{-1})\mathcal{N}(0, \sigma_{\dot{\phi}}^2)((\phi_0 - \phi_{-1})/T)$ with $\sigma_\phi = .2$ and $\sigma_{\dot{\phi}} = 2$. In the first case (Figure 2a), where the bound is tight, $\sigma_w = 15$ and the expectations were computed by summing over 10^3 state trajectories. This case is weakly nonlinear since the sample mean and standard deviation of $\max_{k=-1,\ldots,500} |\phi_k|$ is 0.758271 ± 0.271391. In the second case (Figure 2b), where the bound is loose, $\sigma_w = 20$ and the expectations were computed by summing over 10^5 state trajectories. This case is moderately nonlinear since the sample mean and standard deviation of $\max_{k=-1,\ldots,500} |\phi_k|$ is 0.97351 ± 0.363154. With these parameters the EKF makes rare but large errors analogous to cycle slips in a phase-locked loop so a larger number of trajectories were used to evaluate its performance than in the first case. Finally, in the third case, the system is very nonlinear and the SNR is poor. This problem, modeled after acquisition in a phase locked loop, has a nearly uniformly distributed initial condition on the angle and a large variance process noise. Specifically, $\sigma_w = 25$ and the pdf on the initial condition is $p(\phi_{-1}, \phi_0) = [U(\pi) * \mathcal{N}(0, \sigma_p^2)](\phi_{-1})\mathcal{N}(0, \sigma_{\dot{\phi}}^2)((\phi_0 - \phi_{-1})/T)$ where $U(x)$ is the uniform distribution on the interval $[-x, +x]$, $*$ is convolution, and $\sigma_p = .5$ and $\sigma_{\dot{\phi}} = 2$. This case is strongly nonlinear since the sample mean and standard deviation of $\max_{k=-1,\ldots,500} |\phi_k|$ is 2.55681 ± 2.21283. Though the EKF performance is poor and is not shown, the CRB can still be computed without difficulty, as shown in Figure 2c based on 10^3 state trajectories.

3 Conclusions

A Cramer-Rao bound for the mean squared error that can be achieved with nonlinear observations of a nonlinear p-th order AR process where both the process and observation noise covariances can be state dependent is presented. The major limitation is that the AR process must be driven by an additive white Gaussian noise process that has a full-rank covariance. The bound is a generalization of the results of Refs. [4, 2] to the case $p > 1$ and state-dependent noises. In addition, its computation requires different methods, specifically the solution of a system of quadratic equations for which a recursive method is described.

Relaxation of the full-rank condition on the process noise covariance will probably require constrained CRB tools [8, 5]. The merger of the constrained CRB tools with the dynamical system approach of this paper and Refs. [4, 2] does not appear to be straightforward.

References

[1] J. S. Abel. A bound on mean-square-estimate error. *IEEE Trans. Info. Theory*, 39(5):1675–1680, September 1993.

[2] B. Z. Bobrovsky and M. Zakai. A lower bound on the estimation error for Markov processes. *IEEE Trans. Auto. Contr.*, 20(6):785–788, December 1975.

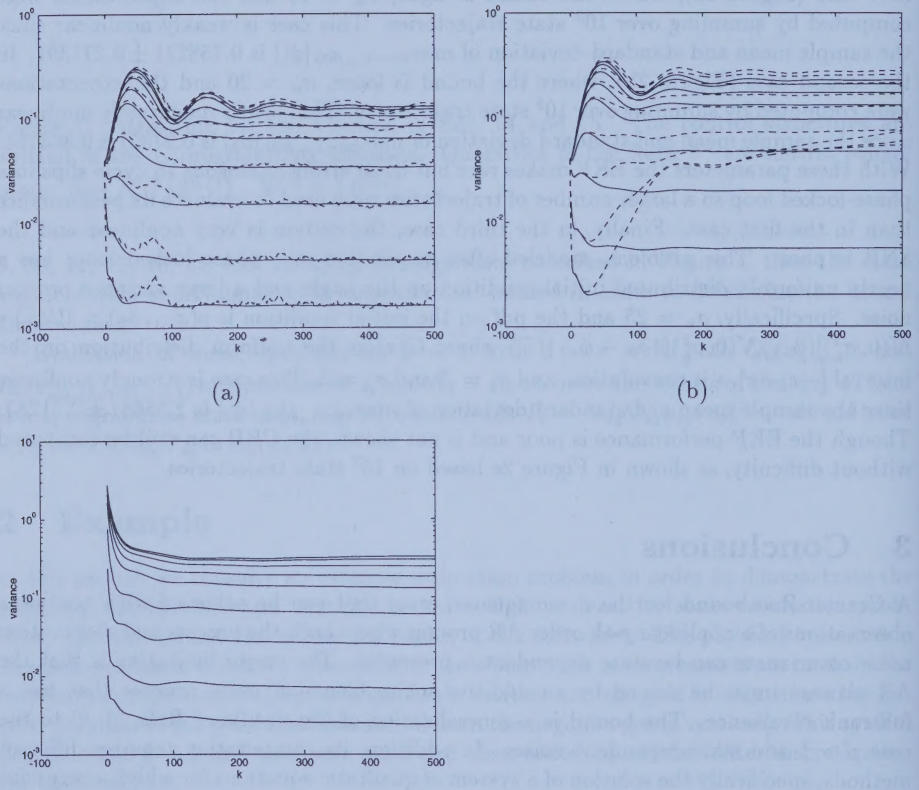


Figure 2: Comparison of CRB (solid line) and EKF estimation variance (dotted line) for a range of measurement noise variance: $\sigma_v \in \{.1, .2, .5, 1, 2, 3, 5, 10, 100\}$. (a) Gaussian initial condition, $\sigma_w = 15$. (b) Gaussian initial condition, $\sigma_w = 20$. (c) Nearly uniform initial condition, $\sigma_w = 25$.

[3] B. Z. Bobrovsky and M. Zakai. Comments on "Status of CR-like lower bounds for nonlinear filtering". *IEEE Trans. Aero. Elect. Syst.*, 26(5):895–896, September 1990.

[4] J. I. Galdos. A Cramer-Rao bound for multidimensional discrete-time dynamical systems. *IEEE Trans. Auto. Contr.*, 25(1):117–119, February 1980.

[5] J. D. Gorman and A. O. Hero. Lower bounds for parametric estimation with constraints. *IEEE Trans. Info. Theory*, 36(6):1285–1301, November 1990.

[6] T. H. Kerr. Status of CR-like lower bounds for nonlinear filtering. *IEEE Trans. Aero. Elect. Syst.*, 25(5):590–601, September 1989.

[7] T. H. Kerr. Author's reply to comments on "Status of CR-like lower bounds for nonlinear filtering". *IEEE Trans. Aero. Elect. Syst.*, 26(5):896–898, September 1990.

[8] T. L. Marzetta. A simple derivation of the constrained multiple parameter Cramer-Rao bound. *IEEE Trans. Sig. Proc.*, 41(6):2247–2249, June 1993.

A Feature Selection Algorithm for Human Signature Verification*

Yi-Jen Chiu and Toby Berger
School of Electrical Engineering
Cornell University
Ithaca, New York 14853
yjchiu/berger@ee.cornell.edu

Abstract

This paper describes a verification system for human signatures written on a pen-input graphics tablet. The verification algorithm combines feature selection, classification and performance analysis phases to construct customized signature verifiers that employ a small number of features for each individual selected on the basis of training data consisting of only a modest number of genuine signatures.

1 Introduction

Signature verification has become an active area of biometry[6],[7] because it promises automation of the job of verifying signatures in the commercial, financial and computing communities. The design and implementation of a signature verification system faces the following problems: data preprocessing, feature extraction, feature selection, classification and performance analysis.

This paper is concerned mainly with the feature selection component of signature verifier design. By this we mean the problem of finding a small but effective subset of features based on training data that consist of only a modest number of genuine signatures. The desired end is a reliable on-line human signature verification system for point-of-sale and

*This work was supported in part by the National Science Foundation under grants NCR-8903288, NCR-9216975 and IRI-9310670

point-of-delivery applications.

In mathematical parlance, feature selection is known as dimensionality reduction, a mapping of a high-dimensional vector representation of the original signals into a small but effective subset of features. Because many features are not linear functions of the original signals, feature selection does not reduce to finding the coefficients of a linear function that optimizes a performance criterion.

The Bayes likelihood ratio classifier is known to be optimal in the sense of minimizing the risk function[4]. However, in order to get the likelihood ratio, we need to have extensive training data to estimate the conditional probability density function. In many applications, including signature verification, it is not possible to get enough training data.

Actually, the success of a signature verifier, or for that matter of any pattern recognition system, depends both on the classifier design and on the features selected for the classifier to operate upon[3]. We propose a pretesting algorithm that selects features by combining the feature selection phase with the classification and performance analysis phases.

We attack the availability of the limited number of training data by proposing a leave-half-out procedure to simulate more testing cases. A nonliner classifier, called the majority classifier, is included in our pretesting process. We use the False Rejection Rate(FRR, TYPE I error rate) and False Acceptance Rate(FAR, TYPE II error rate) directly as the performance criterion because low FRR and FAR are indeed the ultimate objective in signature verification system design.

Other feature selection algorithms on different classifiers have also been implemented in this work. Our simulation results show that the leave-half-out procedure combined with the majority classifier is the most competitive.

2 Data Collection and Feature Extraction

Data collection and feature extraction encompass all the processes that intervene between when a person signs until we get the candidate features of this signature. A brief description is given here; for a more detailed

one, see Lee[5].

2.1 Data Acquisition

Our physical signature verification system consists of a graphics tablet interfaced with a personal computer. The graphics tablet communicates with the PC through a standard RS232 serial communication in asynchronous mode at 9600 baud. The tablet outputs the X, Y coordinates of the position of the pen to the PC. We designed an assembler program which establishes an appropriate communication channel between them, and simultaneously programs the timer chip in the PC in order to extract timing information about the written signature.

2.2 Data base

We have performed our experiment on a total of 1650 genuine signatures which were collected from a population of 19 human subjects; 50 or 100 signatures from each subject. We also have collected a total of 608 forgeries; 8 forgers for each genuine signer with 4 forgeries from each forger. So, for each subject, we have 50 or 100 genuine signatures and 32 forgeries.

2.3 Feature Extraction

A collection of 62 parameter features has been employed in this work as the candidate feature set. "Parameter feature" here means a numerical quantity computed from the measurement; that is, from the X, Y coordinates and the timing information of the pen movement. This 62-feature set contains not only static information such as the length, area,... but also dynamic information such as the speed, acceleration,... We normalized most of these 62 candidate parameter features because, from preliminary study[2], we found that human signature verification is quite sensitive to variations in size and speed of genuine signatures.

3 Feature Selection

Our aim is to find a small but effective feature set for each individual. The motivation is both to reduce the complexity of computation and to capture maximally the distinctive information contained in the signatures based on a modest amount of training data. A different feature set

probably would result for each individual if we were to use a different feature selection scheme and/or a different classifier structure. We compare three feature selection algorithms based on randomly choosing 10 genuine signatures and 8 forgeries from each subject as the training data. We find that the performance of appropriately individualized 3-feature sets is acceptable.

3.1 Leave-Half-Out(LHO) Feature Selection

The LHO is a pretesting algorithm that explores the feature selection task by choosing the feature set that performs best in the pretesting cases based on the training data consisting of a modest number of genuine signatures.

Suppose, in each signer's reference database, we have n genuine signatures, denoted as a group by T_{ref}, and m forgeries, denoted as F_{ref}. First we divide T_{ref} into two groups; half of them, say T_{est}, for estimating the statistics(for example, the mean and the standard deviation) of the parameter features and the other half of them, T_{test}, for pretesting. We switch one member in T_{est} and T_{test} after each pretesting case. Therefore, for each subject, we have for each candidate set of features $t = \binom{n}{\frac{n}{2}}$ pretesting cases involving a total of $t*n/2$ overlapping pretesting genuine signatures and $t * m$ overlapping pretesting forgeries.

In each pretesting case, we perform the tests on each member in T_{test} and F_{ref}. Instead of sticking to one threshold value, we perform tests on several threshold values, say k of them. Thus, for each feature combination, we perform

$$(m + n/2) * k * \binom{n}{\frac{n}{2}} \tag{1}$$

pretests. Eventually, for each subject we choose the triplet feature set having the best pretesting FRR-FAR tradeoff over the range of the thresholds. The flow chart of the LHO algorithm is illustrated in Fig. 1.

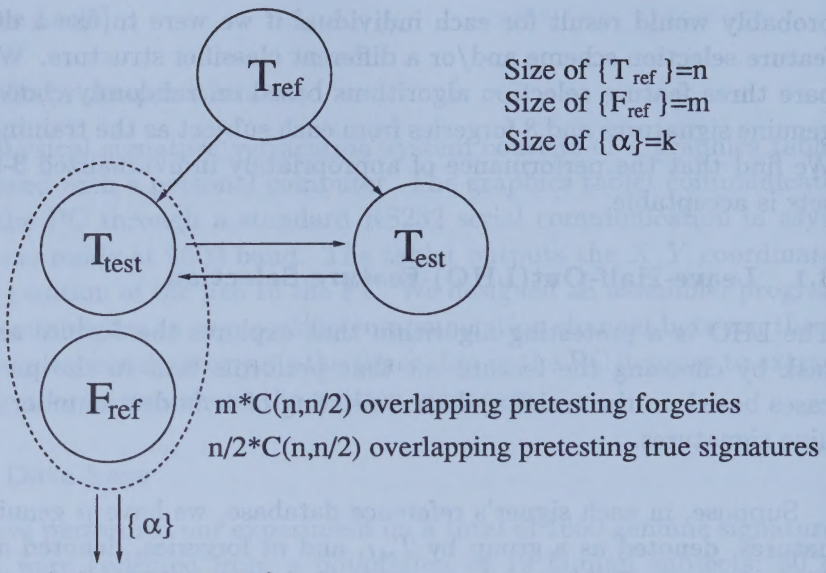


Figure 1: Leave-Half-Out feature selection algorithm

3.2 Fisher Discriminant Analysis(FDA) Feature Selection

We define the Fisher linear discriminant distance measure

$$d_i = \frac{|u(t,i) - u(f,i)|}{\sqrt{\sigma^2(t,i) + \sigma^2(f,i)}} \tag{2}$$

between the classes of genuine and forgery signatures for feature i, with arguments t, f denoting the genuine and forgery classes respectively. $u(t,i)$, $u(f,i)$, $\sigma(t,i)$, and $\sigma(f,i)$ are, respectively, the sample means and the sample covariances of feature i computed from the data bases of genuine and forgery classes.

When forgery data are available, feature selection is carried out by comparing the genuine signatures to the signatures in the forgery data base. Feature i is considered more important than feature j if

$$d_i > d_j \tag{3}$$

FDA analysis selects the features with the largest $d_i's$.

3.3 Principal Component Analysis(PCA) Feature Selection

Let X be an $n-$dimensional feature vector. Then X can be represented by an expansion of the form

$$X = \sum_{j=1}^{n} y_i \phi_i = \Phi_i Y \tag{4}$$

where

$$\Phi = [\phi_1 .... \phi_n] \tag{5}$$

and

$$Y = [y_1 .... y_n]^T \tag{6}$$

The deterministic matrix Φ is made up of n linearly independent column vectors. Without loss of the generality, we assume the columns of Φ form an orthonormal set. Then the components of Y are given by

$$y_i = \phi_i^T X \qquad (i = 1, ..., n) \tag{7}$$

Thus, Y results from applying an orthonormal transformation to the feature vector X, and is itself a random vector. For purposes of feature selection, we want to determine only $m(< n)$ components of Y that still characterize X satisfactorily. The optimal such representation in the minimum mean-square error sense, call it $\tilde{Y}$, is obtained by selecting the $\phi_i' s$ as the eigenvectors corresponding to the largest m eigenvalues of the covariance matrix of X[8]. Said $\tilde{Y}$ is our PCA feature vector.

4 Classifier Design

The following is an introduction to the majority classifier and the Euclidean classifier.

4.1 Majority(MAJ) Classifier

A majority classifier declares that the signature being tested is genuine if half or more of the tested features' values exceed pre-determined thresholds. Otherwise, the majority classifier declares the signature being tested to be a forgery.

Let u_i and σ_i represent the sample average and sample standard deviation of feature i, respectively. Let n denote the total number of features

used for decision. Consider the following event for the value x_i of feature i:

$$\varepsilon_i = \frac{|x_i - u_i|}{\sigma_i} \leq \alpha, \tag{8}$$

where α is the decision threshold used in the majority classifier. For each individual feature, say feature i, the δ-function test is carried on event ε_i,

$$\delta(\varepsilon_i) = \begin{cases} 1 & \text{if event } \varepsilon_i \text{ holds} \\ 0 & \text{otherwise} \end{cases}$$

where $i = 1, 2, ..., n$. The overall majority decision rule therefore is

$$\sum_{i=1}^{n} \delta(\varepsilon_i) \begin{cases} \geq n/2, & \text{then T is declared a genuine signature} \\ < n/2, & \text{then T is declared a forgery} \end{cases}$$

4.2 Euclidean(EUC) Distance Classifier

The Euclidean distance measure is

$$G(T) = \frac{1}{n} \sum_{j=1}^{n} |\frac{x_i - u_i}{\sigma_i}|^2, \tag{9}$$

where the signature T being tested has the feature vector X and u_i and σ_i are, respectively, the i^{th} feature's sample mean and sample standard deviation for the genuine signatures.

5 Performance Evaluation

5.1 Simulation schemes

We endeavor to find the best 3-feature set for each subject by randomly choosing 10 genuine signatures and 8 forgeries from each subject to serve as the training data set for the simulation of each decision scheme. Here, we present simulation results for four decision schemes:

LHOMAJ: (LHO feature selection and MAJ classification)
FDAMAJ: (FDA feature selection and MAJ classification)
PCAMAJ: (PCA feature selection and MAJ classification)
PCAEUC: (PCA feature selection and EUC classification)

5.2 Results

The results shown in Fig. 2 and Fig. 3 are the average performances over 19 subjects for different simulation schemes. Fig. 2 shows the TYPE II

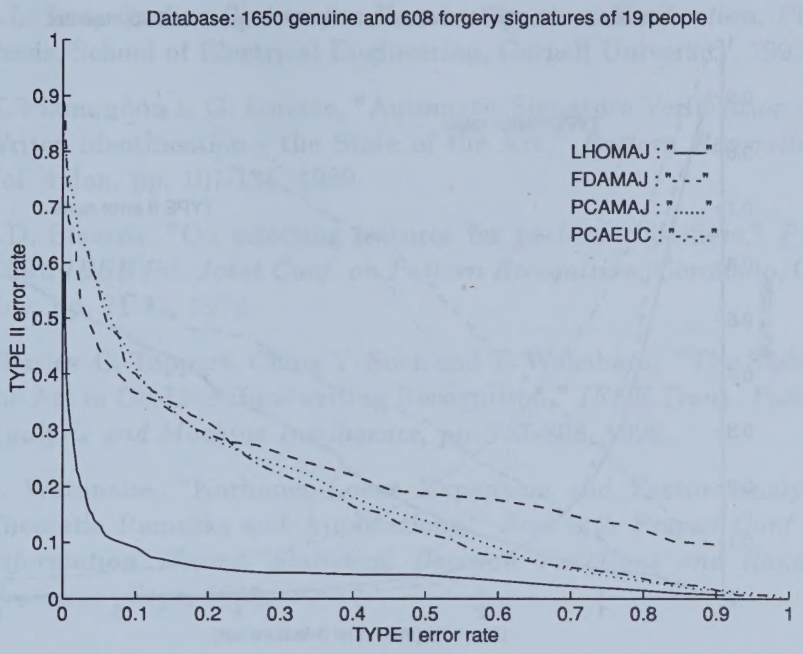


Figure 2: Performance of several simulation schemes over some thresholds

vs TYPE I error rates. The LHOMAJ scheme performs best, obtaining equal 91% detection rates for the genuine signatures and the forgeries. Fig. 3 gives us the operating threshold information for the different TYPE I and TYPE II error rates. The LHOMAJ scheme trained with $k = 10$ thresholds spaced between 1.0 and 2.0 achieves; the equal TYPE I/TYPE II error rate point occurs when the threshold equals 1.7. Perhaps more importantly, it also achieves 70% forgery detection rate at a genuine acceptance rate of 99% for a threshold of 3.6, which constitutes satisfactory performance for point-of-sale and point-of-delivery systems. Had we trained with thresholds between 3.0 and 4.0 instead of between 1.0 and 2.0, we might have obtained still better performance for such sale and delivery applications and might have selected quite different feature sets.

6 Conclusions

We sought to design a signature verification system under the practical constraint of availability of only a modest number of reference signatures.

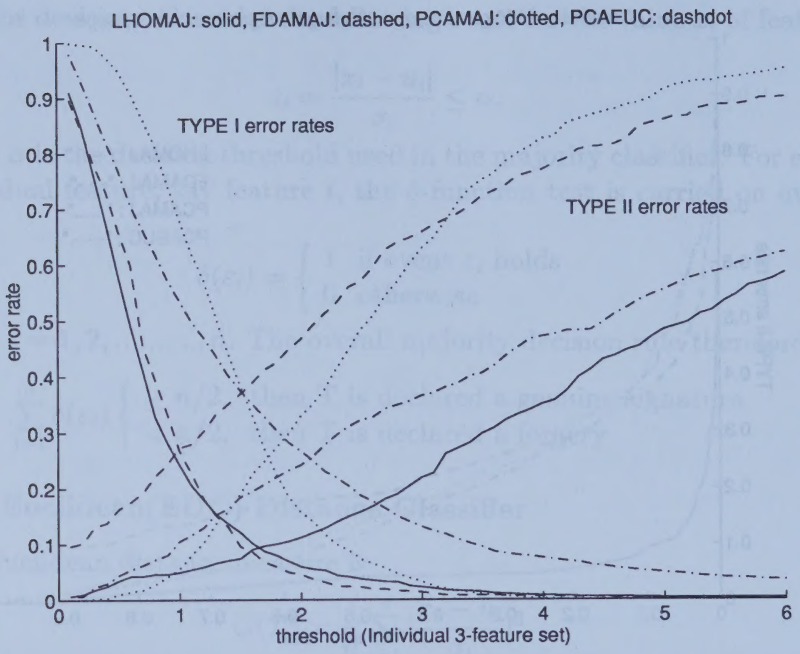


Figure 3: TYPE II vs TYPE I error rates for several simulation schemes

Simulations employing real signatures only a few of which were used as the training data revealed that individualized feature selection using the LHOMAJ scheme shows considerable promise for the human signature problem. Theoretical justification of the leave-half-out procedure is still under investigation.

References

[1] M. Achemlal and M. Mourier, "Dynamic Signature Verification," *Security and Protection in Information Systems*, IFIP, pp. 381-389, 1989.

[2] Y.J. Chiu, *On-Line Signature Verification*, M-Eng Report, School of Electrical Engineering, Cornell University, Jan. 1992.

[3] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, Wiley_Interscience, New York, 1973.

[4] T.S. Ferguson, *A Decision-Theoretic Approach,* Academic Press, New York, 1967.

[5] L.L. Lee, *On-line System for Human Signature Verification*, Ph.D thesis, School of Electrical Engineering, Cornell University, 1992.

[6] R. Plamondon & G. Lorette, "Automatic Signature Verification and Writer Identification - the State of the Art," *Pattern Recognition*, Vol. 4 Jan. pp. 107-131, 1989.

[7] S.D. Stearns, "On selecting features for pattern classifiers," *Proc. Third IEEE Int. Joint Conf. on Pattern Recognition*, Coronado, CA. Nov. pp. 71-75, 1976.

[8] Charles .C. Tappert, Ching Y. Suen and T. Wakahara, "The State of the Art in On-Line Handwriting Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, pp. 787-808, 1990.

[9] S. Watanabe, "Karhunen-Loeve Expansion and Factor Analysis: Theoretic Remarks and Applications," *Proc. 4th Prague Conf. on Information Theory, Statistical Decision Functions and Random Processes* , Sept. 1965.

A Robust Order Statistic Estimator with Rapid Adaptation [1]

PETER M. CLARKSON and QI FAN

Biomedical Engineering Center
The Ohio State University
Columbus, OH 43210

Abstract

For signal estimation problems, one may formulate optimal, *minimum variance unbiased* (MVUB) *order statistic* OS estimators, which are typically much more efficient than comparable linear operators. However, the form of the MVUB OS estimators involves the autocorrelation matrix of the ordered samples, which in turn implies prior knowledge of the noise distribution, and demands significant computation. Moreover, while gradient-based the MVUB OS estimator can be formulated, adaptation is typically slow. We propose an OS estimator that is guaranteed to be robust, and is capable of rapid adaptation, while providing close to optimal efficiency. The algorithm comprises a two-stage estimator with outlier detection and removal followed by adaptive trimmed mean filtering. We demonstrate that this estimator is β-robust and has the highest possible breakdown point. Using a range of data, simulation studies exhibit that the mean-squared error for the algorithm is close to that for the fixed optimum filter but without prior knowledge of the distribution.

1 Introduction

Order statistic (OS) operations form the basis for a powerful class of nonlinear digital filters [1]-[3]. Several OS filters, most notably the median filter, have been widely applied in signal and image processing. In signal estimation problems, one may formulate optimal, *minimum variance unbiased* (MVUB) OS estimators whose statistical efficiency generally exceeds that of linear estimators. However, the form of the MVUB OS estimators involves the autocorrelation matrix of the ordered samples, which in turn implies prior knowledge of the noise distribution, and demands significant computation.

With the high level of interest in OS filters, a natural development has been the design of algorithms to exploit the powerful properties of OS operators in an adaptive framework. A class of gradient-based *adaptive order statistic filters* (AOSFs) has been proposed [4], [5]. While having the potential to achieve the MVUB solution, and bypass the difficulty of solving the ordered autocorrelation matrix, these algorithms are limited by the slow convergence associated with gradient-descent algorithms. Moreover, these AOSFs may take the form of *any OS* operator and while many OS operators are distributionally robust, others, including the arithmetic mean, are not.

Here, we propose a robust OS estimator which can be easily computed from sample data, and has guaranteed robustness while providing close to optimal statistical efficiency. We analyze the properties of this algorithm and demonstrate that this estimator is β - robust and has the highest possible breakdown point. We use *mean-squared error* (MSE) comparisons to examine the efficiency of the estimator, and illustrate that this estimator

[1]This work was supported by the NATIONAL SCIENCE FOUNDATION under grant no. MIP-9220769.

achieves close to the optimal statistical efficiency but without prior knowledge of the noise statistics.

2 The Minimum Variance Unbiased Order Statistic Filter

Consider the following scenario: An unknown, locally constant signal s is embedded in zero-mean stationary noise $\{\eta(n)\}$, producing a measurement sequence $\{x(n)\}$ where

$$x(n) = s + \eta(n), \tag{1}$$

in some neighborhood of n. Consider the problem of estimating s from the observations. This problem may be approached via the application of *order statistic* (OS) filters to the noisy measurements $x(n)$. We can define an OS filter as follows: An N point non-recursive OS filter with input $\{x(n)\}$ produces output $\{z(n)\}$ according to the relation

$$z(n) = \boldsymbol{a}^T \boldsymbol{x}_{(n)} \triangleq OS_{\boldsymbol{a}}\{\boldsymbol{x}(n)\}_N, \tag{2}$$

where the components $x_{(n)}(i)$ of the vector $\boldsymbol{x}_{(n)}$ are the last N input values $x(n), \ldots, x(n-N+1)$ ranked as $x_{(1)}(n) \leq x_{(2)}(n) \leq \cdots \leq x_{(N)}(n)$, where $\boldsymbol{a}$ is a vector weighting these ranked values, and where $\sum_{i=1}^{N} a_{(i)} = 1$ and $a(i) = a(N-i)$, $i = 1, 2, ..., (N-1)/2$.

We can construct a *minimum variance unbiased* (MVUB) OS filter as the set of weights $\boldsymbol{a}^*$ that minimizes

$$J = E\{(z(n) - s)^2\}; \text{ subject to } E\{z(n)\} = s, \tag{3}$$

where $z(n)$ is the OS filter output of Eq. (2). Combining Eqs. (2) and (1) we may write

$$E\{z(n)\} = \boldsymbol{a}^T \mathbf{1} s + \boldsymbol{a}^T E\{\boldsymbol{\eta}_{(n)}\}\ , \tag{4}$$

where $\mathbf{1}^T = [1\ 1 \cdots\ 1]$, and $\boldsymbol{\eta}_{(n)}^T = [\eta_{(n)}\ \eta_{(n-1)}\ \cdots\ \eta_{(n-N+1)}]$ is the vector of ranked samples. This constrained minimization problem may be solved using Lagrange multipliers to give

$$\boldsymbol{a}^* = R_{(\eta)}^{-1} C (C^T R_{(\eta)}^{-1} C)^{-1} \mathcal{F}\ , \tag{5}$$

where $R_{(\eta)}$ is the $(N \times N)$ autocorrelation matrix of ordered noise measurements with elements $R_{(\eta)}(i,j) = E\{\eta_{(i)}(n)\eta_{(j)}(n)\}$.[2]

For symmetric iid observations, the MVUB OS estimator $\boldsymbol{a}^*$ is the median when the noise distribution is Laplacian, the average when it is Gaussian, and the outer mean when it is uniform. For iid errors, it can easily be shown that the variance of this MVUB order statistic estimator is less than or equal to the output variance corresponding to the *best linear unbiased estimator*. In fact, for many distributions, the MVUB OS operator corresponds to the *uniformly* minimum variance unbiased estimator among *all* possible estimators [7].

[2]Interestingly, the solution exactly parallels that for the constrained minimum variance linear beamformer [6].

3 Adaptive Order Statistic Filters

Returning to the constant signal model of Eq. (1), as we have seen, the MVUB solution depends on the correlation matrix $R_{(\eta)}$. Bovik *et al.* [2] determined the optimal filter vector $\boldsymbol{a}^*$ using known statistics for the observed noise. On the other hand, it is often necessary to construct OS filters *without* prior knowledge of the noise distribution. One approach is simply to form block estimates from the data which are then used to construct the OS filter [8]. Another possibility is to apply an iterative update to the OS filter vector. We refer to such a system as an *adaptive order statistic filter* (AOSF). Pitas and Venetsanopoulos [4] proposed a gradient based AOSF algorithm intended to achieve the constrained minimization of Eq. (3) using an LMS type update. Crucially, however, their algorithm forces satisfaction of the unbiasedness constraint by scaling the filter coefficients. The resulting filter vector does not converge to $\boldsymbol{a}^*$ in general, and the filter suffers from suboptimal and potentially poor performance. A gradient-based algorithm that *does* satisfy the unbiasedness constraint was proposed by Williamson and Clarkson [5]. This AOSF algorithm uses a projection-based constrained minimization procedure analogous to that of Frost [6], but applied to ordered data. This algorithm produces adaptive updates of the MVUB estimator and *does* produce efficient estimates. On the other hand, the filter suffers from the slow convergence associated with gradient-descent algorithms. Moreover, the AOSF can take the form of *any* OS operator and not all OS operators are robust. Thus, in some applications we must sacrifice the optimality of the MVUB AOSF in order to obtain rapid adaptation and to ensure robustness to outliers.

4 OS Filters and Robustness

We may characterize the robustness of OS estimators in a variety of ways. Two useful measures are the *breakdown point* [9] and the *influence function* [10]:

1. *Breakdown Point* – The breakdown point of an estimator is the smallest fraction of the observations that must be replaced by unbounded outliers before the estimation error can become unbounded. For example, the breakdown point of the average, applied to N observations, is equal to $1/N$ because it is sufficient to replace a single observation by one unbounded value. On the other hand, the sample median possesses the best possible breakdown point, namely $(N+1)/2N$ – we have to replace more than half of the observations by outliers in order for the median to lie among the outliers.

2. *Influence Function* – The influence function describes the effect on an estimate $T(F)$ of an infinitesimal contamination at point x. Formally, let $\hat{s} = T_N(x(1), ..., x(N))$ be an estimator of a parameter s with asymptotic value:

$$\lim_{N\to\infty} \{T_N(x(1), x(2), ..., x(N))\} \to T(G), \tag{6}$$

where $G(x)$ is the distribution of the iid observations $x(i)$. The *influence function* (IF) of T at F is defined by

$$IF(x; T, F) = \lim_{\delta\to 0} \left\{ \frac{T((1-\delta)F + \delta\Delta_x) - T(F)}{\delta} \right\}, \tag{7}$$

where Δ_x is a point mass at x. For example, for zero-mean symmetric distributions, the mean and median have influence functions:

$$IF_{mean}(x;T,\Phi) = x \tag{8}$$

$$IF_{med}(x;T,\Phi) = \frac{sgn(x)}{2f(0)}, \tag{9}$$

where $sgn(\ )$ denotes the sign operator. This clearly illustrates the distinction between the two estimators – for the mean, the IF is unbounded – there is no limit to the influence of a single observation, while for the median the IF is strictly limited.

A useful measure derived from the IF is the *gross error sensitivity* [10]

$$\gamma^* = \sup_x |IF(x;T,F)|. \tag{10}$$

Estimates for which γ^* is finite are called *β-robust.* It is immediately apparent that the median is β-robust, while the mean is not.

In general, for outlier corrupted data a compromise between efficiency and robustness is provided by the trimmed mean operator. The breakdown point of the trimmed mean is $(M+1)/N$. This dependence on the trimming parameter M allows us to generate a continuous range of breakdown values between the extremes of the mean ($M = 0$) and the median ($M = (N-1)/2$). Similarly, the IF of the trimmed mean lies between the extremes of mean and median. For data that is outlier corrupted, we may argue that restricting the OS operator to take the form of a trimmed mean with M adjusted adaptively, provides good, though not optimal efficiency, while giving good resistance to outliers.

5 A Two-Stage Adaptive OS Estimator

The adaptive procedure described in this paper comprises a two-stage estimator with outlier detection and removal followed by adaptive trimmed mean filtering (see Fig. 1).

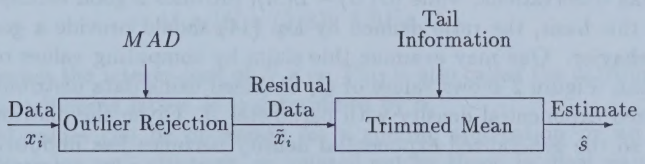


Figure 1. Diagram of estimation scheme

In the first stage, a highly robust outlier detection mechanism is applied to the observations. This both removes outliers from the data, and produces a residual set of observations that are approximately symmetric.

A suitable outlier detection mechanism can be derived from the *median absolute deviation* (MAD), suggested by Andrews *et al.* [11] as a robust estimator of the 'scale' of a set of observations as an alternative to the standard deviation which is notoriously

non-robust. The MAD is the median of the absolute distances from the sample median of the observations and as such is the natural counterpart of the median for the estimation of scale [10]. For samples $x(i)$ for $i = 1, 2, ..., N$, the MAD is defined by

$$MAD = \ median \ |x(i) - x_{med}| \ ; \ i = 1, 2, ..., N, \tag{11}$$

where x_{med} is the sample median of the observations $x(i)$ for $i = 1, 2, ..., N$.

In the first stage of the estimator, an observation $x(i)$, such that

$$|x(i) - x_{med}| \geq K \times MAD, \tag{12}$$

is discarded as an outlier. Here, K is a constant chosen empirically in the range $4.0 - 5.5$.[3]

In the second stage of processing, the residual observations are subjected to a trimmed mean filtering operation, as defined by Eq. (2), with

$$\boldsymbol{a}^T = [\underbrace{0 \cdots 0}_{M} \ \frac{1}{N-2M} \cdots \frac{1}{N-2M} \ \underbrace{0 \cdots 0}_{M}]^T. \tag{13}$$

The trimming factor M is adapted on the basis of a measure of the tail of the distribution of the residual observations. The approach taken here is to use the measure Q_1 suggested by Hogg [12], and more recently adopted by Restrepo and Bovik [8]:

$$Q_1 = \frac{\overline{U}(.2) - \overline{L}(.2)}{\overline{U}(.5) - \overline{L}(.5)}, \tag{14}$$

where

$$\overline{U}(\alpha) = \frac{1}{N\alpha} \sum_{i=N-[N\alpha]+1}^{N} x_{(i)} + (1 - \frac{[N\alpha]}{N\alpha}) x_{(N-[N\alpha])}, \tag{15}$$

$$\overline{L}(\alpha) = \frac{1}{N\alpha} \sum_{i=1}^{[N\alpha]} x_{(i)} + (1 - \frac{[N\alpha]}{N\alpha}) x_{(N-[N\alpha]+1)}. \tag{16}$$

$\overline{U}(\alpha)$ is the average of the $N\alpha$ largest order statistics, and $\overline{L}(\alpha)$ is defined similarly for the smallest order statistics, and where in both cases, $[\cdot]$ denotes the integer part of the argument. Hogg [12] argues that $[\overline{U}(.2) - \overline{L}(.2)]$ gives a good estimate of scale for Gaussian observations, while $[\overline{U}(.5) - \overline{L}(.5)]$ provides a good estimate for Laplacian data. On this basis, the ratio defined by Eq. (14) should provide a good indication of the tail behavior. One may examine this claim by computing values of Q_1 for various distributions. Figure 2 shows values of Q_1 obtained using data distributed according to a generalized exponential density with parameter β. Given the distribution here, as β increases, so the generalized exponential density becomes less impulsive. Referring to Fig. 2 the decrease in Q_1 obtained as β increases is precisely what we would expect. Logically, the trimming factor M in Eq. (13) is then determined by comparing the value Q_1 against a set of predetermined thresholds. When the distribution is more impulsive, the tails are relatively heavy, Q_1 is higher, and the trimming factor is chosen to give a filter that is closer to the median; conversely when the distribution is less impulsive, Q_1 is lower, and the estimator is more like the average. The algorithm proposed here

[3]Sometimes the MAD of Eq. (11) includes a normalizing factor to produce a scale estimate that is consistent with the standard deviation when applied to Gaussian observations. Here, we absorb this factor into the constant K.

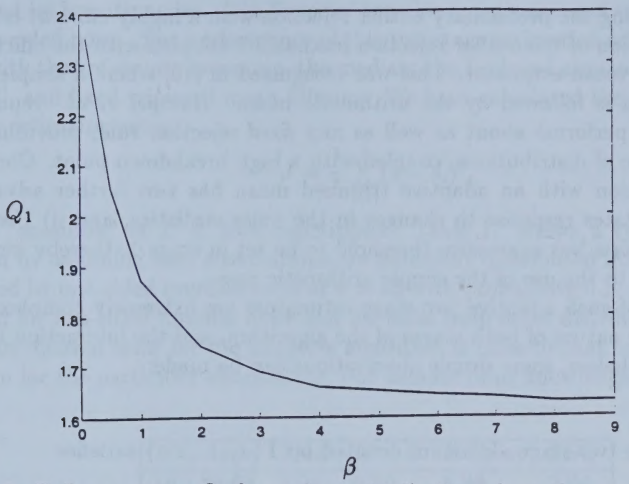


Figure 2: Tail parameter Q_1 for various distributions from the generalized exponential family parameterized by β. Each point is computed by averaging raw Q_1 values over an 'ensemble' of 1000 trials. $N = 20$ in each case.

uses a threshold rule depending on N that is derived from a similar scheme suggested by Hogg [12]. The values we have used are:

$$
\begin{aligned}
10 < N \leq 20 \,; \quad & M = [N \times (1/8)]; \; Q_1 \leq 1.84 \\
; \quad & M = [N \times (2/8)]; \; Q_1 > 1.84 \\
20 < N \leq 35 \,; \quad & M = [N \times (3/32)]; \; Q_1 \leq 1.81 \\
; \quad & M = [N \times (6/32)]; \; 1.81 < Q_1 \leq 1.87 \\
; \quad & M = [N \times (9/32)]; \; 1.87 < Q_1 \\
35 < N \quad ; \quad & M = [N \times (0)]; \; Q_1 \leq 1.75 \\
; \quad & M = [N \times (3/32)]; \; 1.75 < Q_1 \leq 1.81 \\
; \quad & M = [N \times (6/32)]; \; 1.81 < Q_1 \leq 1.87 \\
; \quad & M = [N \times (9/32)]; \; 1.87 < Q_1 \leq 1.91 \\
; \quad & M = [N \times (3/8)]; \; 1.91 < Q_1 \,.
\end{aligned}
$$

Here, $[N \times \alpha]$ denotes the integer part of $N \times \alpha$. This is also called the α trimmed mean operator since the trimming factor M is determined by α.

The threshold values can be optimized for a specific application by an empirical process using a 'training set', although, as pointed out by Hogg, in most situations the performance of the algorithm is *not* critically dependent on the precise values used for the thresholds.

6 Properties of the Two-Stage Estimator

One may ask what are the benefits of a two-stage estimator such as that described above compared to, say, simple outlier rejection? It is known that while subjective or objective outlier rejection can help prevent disaster due to extreme errors, outlier rejection alone is associated with limited efficiency [10]. As Hampel *et al.* point out, the basic weakness of all hard rejection rules is the inability to cope with contamination which lies in the 'region

of doubt'. Augmenting the preliminary outlier rejection with a highly efficient estimator provides the protection of the outlier rejection mechanism coupled with the efficiency of a mean or trimmed mean estimator. This was recognized in [10] where a simple outlier rejection mechanism is followed by the arithmetic mean. Hampel *et al.* report that this simple scheme performs about as well as any *fixed* rejection rule, providing good efficiency for a range of distributions, coupled with a high breakdown point. Combining the threshold rejection with an adaptive trimmed mean has two further advantages: i) Adaptation facilitates response to changes in the noise statistics, and ii) Use of the trimmed mean allows a less aggressive threshold to be set in stage 1, thereby increasing efficiency compared to the use of the simple arithmetic mean.

The properties of such adaptive two-stage estimators are extremely complex due to the highly nonlinear nature of both stages of the algorithm, and the interaction between the sections. Nevertheless, some simple observations can be made:

1. *Equivariance:*

   The composite two-stage algorithm denoted by $T\{x_1, ..., x_N\}$ satisfies

   $$T\{cx_1 + d, ..., cx_N + d\} = cT\{x_1, ..., x_N\} + d \quad ; \; c > 0, \; -\infty < d < \infty \; . \qquad (17)$$

   This is important, ensuring that simple scaling of the observations or the addition of a *d.c.* offset cannot adversely impact the estimation procedure.

2. *Robustness:*

   The two-stage operator $T\{x_1, x_2, ..., x_N\}$ achieves the highest possible breakdown point $-\ (N+1)/2N$. As measured by the breakdown point, it is as robust as the sample median.

   For the usual adaptive trimmed mean estimator, the breakdown point is $(M+1)/N$, the robustness changes with the trimming factor M. In many cases there exists a trade-off between robustness and efficiency. For the two-stage estimator described above, the first stage uses the MAD to remove the distant outliers. Since it never "breaks down" unless the sample median is an unbounded outlier, so the two-stage operator $T\{x_1, x_2, ..., x_N\}$ achieves the same breakdown point as the sample median, which is the highest possible. It is intuitively obvious that the limiting effect of the outlier rejection mechanism results in a finite influence function. Hence, the two-stage estimator is also β-robust.

3. *Symmetry:*

   The MAD scheme improves the symmetry of the observations by removing the (generally asymmetric) outliers. This is important because for symmetric distributions, Q_1 is independent of the parameter s.

   This result may be viewed as a direct deduction from 2. above. If the underlying distribution $f(v)$ is symmetric, then up to $(N+1)/2N$ distant one-sided outliers will be removed.

7 Simulations

Three kinds of simulated data corresponding to Laplacian, Gaussian, and Uniform distributions have been analyzed. In each case, the data generated were iid, and were additively

corrupted by impulse noise. Two forms of impulse noise were used corresponding to one and two-sided noise. The performance of this two-stage estimation scheme has been compared with that of simple averaging, the median, the Hodges-Lehmann $(H-L)$ estimate[4] [13], [14], and fixed trimmed mean filtering. We have calculated the mean-squared error for each estimator using

$$J = \frac{1}{L}\sum_{i=1}^{L}(s - \hat{s}_i)^2, \tag{18}$$

over an 'ensemble' of $L = 1000$ repetitions. Table 1. shows a typical set of results obtained by assuming that observations are uniformly distributed from 65 to 75, and are corrupted by one-sided impulse noise at $x = 35$ with probability 0.1. Similar results were obtained for two-sided impulse noise and for noise from other distributions. In each case the mean-squared error for the adaptive algorithm is close to that achieved by the fixed optimum for the particular distribution, but *without* prior knowledge of the distribution.

N	Avg	Med	$H-L$	α-$Trim$	TSO
10	24.09	2.94	6.51	6.29	2.94
15	20.19	2.70	4.49	3.42	.92
20	18.19	1.59	2.81	1.74	.73
25	17.08	1.36	1.68	1.23	.49
30	16.01	1.20	2.19	1.14	.45
35	15.85	1.11	1.19	.99	.34
40	15.31	1.00	.91	.81	.25

Table 1: Mean-squared estimation errors obtained for x_k distributed as $f(x) = 0.1\delta(x - 35) + 0.9\ Uniform(65, 75)$. Avg = sample average, Med = sample median, $H - L$ = Hodges-Lehmann estimate, $\alpha - Trim = \alpha$ trimmed mean with $\alpha = .25$, TSO = two-stage operator.

8 Conclusion

Although MVUB OS estimators are generally more efficient than comparable linear operators, the form of the MVUB OS estimators involves the autocorrelation matrix of the ordered samples, which in turn implies prior knowledge of the noise distribution, and demands significant computation. The AOSFs proposed in reference [5] can asymptotically achieve the optimal MVUB OS estimator and by-pass the difficulty of having to obtain the autocorrelation matrix for the ordered data. However, as with all gradient-descent algorithms, the convergence time is significant. In any case, the MVUB OS estimator is not necessarily robust since it may take the form of any OS operator, including non-robust forms such as the sample mean and the outer mean. The estimator proposed here is designed to adapt rapidly. It is as robust as the median operator, having the highest possible breakdown point. Since the IF of the proposed estimator is bounded, it is also β-robust. Simulations have demonstrated that the operator has mean-squared

[4]The estimate is the median of $(x_i + x_j)/2$ for any $i \leq j$. Bickel [15] recommends The $H - L$ estimate in situations where the degree of contamination and type of distribution is not known with great precision.

error comparable to that of the fixed optimal estimator, but without prior knowledge of the data distribution.

References

[1] J.W.Tukey, *Exploratory Data Analysis*, Addison-Wesley, 1971.

[2] A.C.Bovik, T.S.Huang, and D.C.Munson, "A generalization of median filtering using linear combinations of order statistics," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 1342-1349, 1983.

[3] H.G.Longbotham and A.C.Bovik, "Theory of order statistic filters and their relationship to linear filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-37, pp. 275-287, 1989.

[4] I.Pitas and A.N.Venetsanopoulos, "Adaptive filters based on order statistics," *IEEE Trans. Signal Processing*, vol. SP-39, pp. 518-522, 1991.

[5] G.A.Williamson and P.M.Clarkson, "On signal recovery with adaptive order statistic filters," *IEEE Trans. Signal Processing*, vol. SP-40, pp. 2623-2626, 1992.

[6] O.L.Frost, III, "An algorithm for linearly constrained adaptive array processing," *Proc. IEEE*, vol. 60, pp.926-935, 1972.

[7] E.L.Lehmann, *Theory of Point Estimation*, Wiley, 1983.

[8] A.Restrepo and A.Bovik, "Adaptive trimmed-mean filters for image restoration," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-36, pp. 1326-1337, 1988.

[9] F.R.Hample, "A general qualitative definition of robustness," *Ann. Math. Stat.*, vol.42, pp. 1887-1896, 1971.

[10] F.R.Hampel, E.M.Ronchetti, P.J.Rousseeuw and W.A.Stahel, *Robust Statistics: The Approach Based on Influence Functions*, Wiley, 1986.

[11] D.F.Andrews, P.J.Bickel, F.R.Hampel, P.J.Huber, W.H.Rogers and J.W.Tukey, *Robust Estimates of Location: Survey and Advances*, Princeton University Press, Princeton, NJ.

[12] R.V.Hogg, "Adaptive robust procedures: a partial review and some suggestions for future applications and theory," *J. Amer. Statist. Ass.*, vol. 69, pp. 909-927, 1974.

[13] J.L.Hodges Jr. and E.L.Lehmann, "Efficiency of some nonparametric competitors of the t-test," *Ann. Math. Stat.*, vol. 27, pp. 324-325, 1956.

[14] J.L.Hodges Jr. and E.L.Lehmann, "Estimates of location based on rank tests," *Ann. Math. Stat.*, vol. 34, pp. 598-611, 1963.

[15] P.J.Bickel, "On some robust estimates of location," *Ann. Math. Stat.*, vol. 36, pp. 847-858, 1965.

On the Application of the EM Algorithm to Sequence Estimation for Degraded Channels [1]

Costas N. Georghiades and Jae Choong Han
Electrical Engineering Department
Texas A&M University
College Station, TX 77843-3128

Abstract- The expectation-maximization (EM) algorithm was first introduced in the statistics literature as an iterative procedure that under some conditions produces maximum likelihood (ML) parameter estimates. In this paper we investigate the use of the EM algorithm to the problem of sequence estimation in the presence of random disturbances and additive white Gaussian noise. Example applications were the EM-based algorithms provide ML sequence estimates are presented for the random-phase and mobile radio fading channels.

1 Introduction

Consider the classical problem of estimating data sequences $\mathbf{s}$ from observations of received data $\mathbf{r}$,

$$\mathbf{r} = \mathcal{F}(\mathbf{s}, \mathbf{z}) + \mathbf{n},$$

where $\mathbf{n}$ is an additive noise vector (later assumed white and Gaussian), and $\mathcal{F}(\mathbf{s}, \mathbf{z})$ is some function of the desired data sequence $\mathbf{s}$ and a noise vector $\mathbf{z}$ (assumed independent of $\mathbf{n}$). A maximum-likelihood (ML) receiver then performs

$$\max_{\mathbf{s}} \; p(\mathbf{r}|\mathbf{s}) = E_{\mathbf{z}}[p(\mathbf{r}|\mathbf{s}, \mathbf{z})].$$

Obtaining optimum sequence estimates thus involves two steps: 1) computing the likelihood function; and, 2) a maximization over the set (possibly constrained when coding is used) of all admissible sequences. When random parameters are involved, evaluation of the likelihood function may require computing an expectation over the joint statistics of the random parameter vector, a task that more often than not is analytically intractable (especially if $\mathbf{z}$ is not an independent vector). Even if the likelihood function can be obtained analytically off line, however, it is invariably a nonlinear function of $\mathbf{s}$, which makes the maximization step (which must be performed in real-time) computationally infeasible. This is especially so for long and/or constrained sequences. Examples of channels that fit the above model include random amplitude, phase, or frequency channels, and channels with combinations of these disturbances, fading channels, random intersymbol-interference (ISI) channels, *etc.*. For all of the above channels, the problem of optimum sequence estimation is known to be difficult or intractable for the reasons mentioned above. In such cases, an iterative formulation of the sequence estimation problem based on the expectation-maximization (EM) algorithm, originally introduced in the statistics literature, can provide an implementable solution. The EM algorithm had been known to statisticians for a long time, but was made better known to others by

[1]This work was supported by the National Science Foundation under Grant NCR-9016329.

the work of Demster, Laird and Rubin [1] and Wu [2]. The interested reader is strongly advised to read these two references on the general formulation of the EM algorithm and its convergence properties. Outside the statistics literature, the EM algorithm was used by Snyder [3] to obtain ML estimates of radioactivity concentration in positron emission tomography, and by Feder [4] to extract parameter estimates from superimposed signals. For previous applications of the EM algorithm to communication systems see Poor [5], Kaleh [6], Han [7], Georghiades et al. [8, 9, 10], Zabin et al. [11], Ansari et al. [12], and Modestino [13].

We introduce the algorithm briefly in Section 2, followed by Sections 3 and 4 that deal with the application of the EM algorithm to the random phase and fading channels respectively; Section 5 concludes.

2 The EM Algorithm

Let $\mathbf{b} \in \mathcal{B}$ be a set of parameters to be estimated from some observed data $\mathbf{y} \in \mathcal{Y}$. Then the ML estimate $\hat{\mathbf{b}}$ of $\mathbf{b}$ is a solution to

$$\hat{\mathbf{b}} = \arg \max_{\mathbf{b} \in \mathcal{B}} g(\mathbf{y}|\mathbf{b}), \tag{1}$$

where $g(\mathbf{y}|\mathbf{b})$ is the conditional density of the data given the parameter vector to be estimated. As discussed above, in many cases an explicit expression for this conditional density does not exist, or is hard to obtain. In other cases such an expression may exist, but it is one that does not lend itself to efficient maximization over the set of parameters. In such situations and under some conditions, the expectation-maximization algorithm may provide a solution, albeit iterative (and possibly numerical), to the ML estimation problem.

The EM algorithm based solution proceeds as follows. Suppose that instead of the data $\mathbf{y}$ actually available one had data $\mathbf{x} \in \mathcal{X}$ from which $\mathbf{y}$ could be obtained through a many-to-one mapping $\mathbf{x} \rightarrow \mathbf{y}(\mathbf{x})$, and such that their knowledge makes the estimation problem easy (for example, the conditional density $f(\mathbf{x}|\mathbf{b})$ is easily obtained.) The two sets of data $\mathbf{y}$ and $\mathbf{x}$ are referred to as the *incomplete* and *complete* data respectively. The EM algorithm makes use of the log-likelihood function for the complete data in a two-step iterative procedure which under some conditions converges to the ML estimate given in (1) [1, 2]. At each step of the EM iteration, the likelihood-function can be shown to be nondecreasing [1, 2]; if it is also bounded (which is mostly the case in practice), then the algorithm converges. The two-step procedure at the i-th iteration is:

1. E-step: Compute $Q(\mathbf{b}|\mathbf{b}^i) \equiv E[\log f(\mathbf{x}|\mathbf{b})|\mathbf{y}, \mathbf{b}^i]$,
2. M-step: Solve $\mathbf{b}^{i+1} = \arg \max_{\mathbf{b} \in \mathcal{B}} Q(\mathbf{b}|\mathbf{b}^i)$,

where $\mathbf{b}^i$ is the parameter vector estimate at the i-th iteration. The two steps of the iteration are referred to as the expectation (E-step) and maximization (M-step) steps respectively. Note that in the absence of the data $\mathbf{x}$, which makes $\log f(\mathbf{x}|\mathbf{b})$ a random variable, the algorithm maximizes its conditional expectation instead, given the incomplete data and the most recent estimate of the parameter vector to be estimated.

Let us now specialize the algorithm to channels that are perturbed by (unwanted) random parameters. If we let $\mathbf{z}$ be the unwanted random parameter vector, then an

obvious choice (but by no means the only) for the complete data is $\mathbf{x} = (\mathbf{y}, \mathbf{z})$. In this case, $f(\mathbf{x}|\mathbf{b}) = p(\mathbf{y}|\mathbf{z}, \mathbf{b})p(\mathbf{z})$, where $p(\mathbf{z})$ is the a priori density of $\mathbf{z}$ (we assume that $\mathbf{b}$ and $\mathbf{z}$ are independent). Then

$$Q'(\mathbf{b}|\mathbf{b}^i) = E\left[\log p(\mathbf{y}|\mathbf{z}, \mathbf{b})|\mathbf{y}, \mathbf{b}^i\right] + E\left[\log p(\mathbf{z})|\mathbf{y}, \mathbf{b}^i\right],$$

and since the second term above is not a function of $\mathbf{b}$ (and thus it is a constant as far as the M-step is concerned) it can be dropped to yield

$$Q(\mathbf{b}|\mathbf{b}^i) = E\left[\log p(\mathbf{y}|\mathbf{z}, \mathbf{b})|\mathbf{y}, \mathbf{b}^i\right]. \tag{2}$$

The conditional expectation above is with respect to the conditional density of the random parameter vector $\mathbf{z}$ given the incomplete data $\mathbf{y}$ and assuming that $\mathbf{b} = \mathbf{b}^i$. The conditional density $p(\mathbf{z}|\mathbf{y}, \mathbf{b}^i)$ can be expressed as

$$p(\mathbf{z}|\mathbf{y}, \mathbf{b}^i) = A \cdot p(\mathbf{y}|\mathbf{z}, \mathbf{b}^i)p(\mathbf{z}), \tag{3}$$

where A is some constant not a function of $\mathbf{b}$ that can be evaluated, but need not be. Thus,

$$Q(\mathbf{b}|\mathbf{b}^i) = A \cdot \int_{\mathbf{z}} \log[p(\mathbf{y}|\mathbf{z}, \mathbf{b})]p(\mathbf{y}|\mathbf{z}, \mathbf{b}^i)p(\mathbf{z})d\mathbf{z}. \tag{4}$$

Then a necessary condition for a maximum of $Q(\mathbf{b}|\mathbf{b}^i)$ (assuming differentiability) is

$$\int_{\mathbf{z}} \frac{p(\mathbf{y}|\mathbf{z}, \mathbf{b}^i)}{p(\mathbf{y}|\mathbf{z}, \mathbf{b})} \frac{\partial p(\mathbf{y}|\mathbf{z}, \mathbf{b})}{\partial \mathbf{b}} p(\mathbf{z})d\mathbf{z}|_{\mathbf{b}=\mathbf{b}^{i+1}} = \mathbf{0}.$$

It is easy to see that at convergence, in which case $\mathbf{b} = \mathbf{b}^i = \mathbf{b}^{i+1}$, the condition reduces to

$$\frac{\partial p(\mathbf{y}|\mathbf{b})}{\partial \mathbf{b}}|_{\mathbf{b}=\widehat{\mathbf{b}}_{\mathrm{ML}}} = \mathbf{0},$$

which is the necessary condition satisfied by the ML estimate of $\mathbf{b}$.

In order to facilitate understanding of the EM algorithm, we illustrate it with two simple examples.

Example 1:

Consider the simple problem of estimating the parameter s from received data $r = s + n$ where n is a zero-mean, variance σ_n^2 Gaussian random variable. The ML estimate of s given r is then simply r. Clearly, for this problem there is no need to use the EM algorithm which will only make things more complicated, but we will nevertheless do so to illustrate its use. We view r as the incomplete data and define (somewhat arbitrarily) the complete data to be $\mathbf{x} = (r, w)$ where w is a (artificial) zero-mean, variance σ_w^2 Gaussian random variable, not independent of n; n and w are jointly Gaussian with joint density

$$h(w, n) = \frac{1}{2\pi\sigma_n\sigma_w\sqrt{1-\rho^2}} \exp\left\{-\frac{1}{2(1-\rho^2)}\left[\frac{w^2}{\sigma_w^2} - \frac{2\rho wn}{\sigma_n\sigma_w} + \frac{n^2}{\sigma_n^2}\right]\right\}.$$

Then $f(\mathbf{x}|s) = h(w, r-s)$ and the log-likelihood function of the complete data (dropping unnecessary terms and rearranging slightly) is $[s^2 - 2s(r - \rho w\sigma_n/\sigma_w)]$. It follows

$$\begin{aligned} Q(s|s^i) &= E\left[[s^2 - 2s(r - \rho w\sigma_n/\sigma_w)]|r, s^i\right] \\ &= s^2 - 2s\left(r - \rho\frac{\sigma_n}{\sigma_w}\hat{w}_i\right), \end{aligned}$$

where $\hat{w}_i = E[w|r, s^i]$, is the conditional mean of w given the incomplete data r and assuming that $s = s^i$. We can easily show that $p(w|r, s^i) = h(w, r - s^i)/p_n(r - s^i)$, which is Gaussian with mean $\hat{w}_i = \rho\sigma_w(r - s^i)/\sigma_n$. Then,

$$Q(s|s^i) = s^2 - 2s[r(1 - \rho^2) + \rho^2 s^i].$$

This completes the E-step of the algorithm. The maximization step (M-step) can be performed analytically (in this case) to yield the recursion $s^{i+1} = \rho^2 s^i + r(1 - \rho^2)$, which has the explicit solution (s^0 is an arbitrary initial estimate)

$$s^i = \rho^{2i} s^0 + r(1 - \rho^{2i}).$$

Thus, irrespective of the initial estimate s^0, the algorithm converges to the true ML estimate $\hat{s} = r$, for any $0 \leq \rho < 1$. □

Example 1 illustrates that the complete data can be chosen quite arbitrarily. In practice, however, an educated choice may result in a simpler solution.

Example 2:

Let the received data be $r = as + n$ where n and a are zero-mean, Gaussian random variables with variances σ_n^2 and σ_a^2 respectively. The log-likelihood function for this problem is easily obtained as

$$\ell(s) = -\frac{r^2}{(s^2\sigma_a^2 + \sigma_n^2)} - \ln[s^2\sigma_a^2 + \sigma_n^2].$$

Maximization over s yields the possible solutions $\hat{s} = \{0, \pm\sqrt{(r^2 - \sigma_n^2)/\sigma_a^2}\}$. Depending on whether $(r^2 - \sigma_n^2)$ is positive or negative one of the above solutions maximizes the likelihood-function and is thus the ML estimate:

$$\hat{s} = \begin{cases} 0 & \text{if } (r^2 - \sigma_n^2) < 0 \\ \pm\sqrt{\frac{r^2 - \sigma_n^2}{\sigma_a^2}} & \text{otherwise.} \end{cases}$$

Note that, as expected, the sign of s cannot be uniquely resolved.

Let us now look at the EM solution. Let the complete data be $\mathbf{x} = (r, a)$. Then it can be easily found that

$$Q(s|s^i) = -\widehat{a_i^2}s^2 + 2r\hat{a}_i s$$

where $\widehat{a_i^2} = E[a^2|r, s^i]$ and $\hat{a}_i = E[a|r, s^i]$. Performing the maximization step we obtain the recursion

$$s^{i+1} = r^2\frac{\hat{a}_i}{\widehat{a_i^2}}.$$

Substituting the conditional first and second moments obtained easily since $p(a|r, s^i)$ is Gaussian, we finally have

$$s^{i+1} = \frac{r^2 s^i}{\sigma_n^2 + \frac{r^2\sigma_a^2(s^i)^2}{\sigma_a^2(s^i)^2 + \sigma_n^2}}.$$

Solving for the steady-state estimate s^∞ we obtain exactly the same solution set as above. Moreover, irrespective of the initial estimate, the iteration converges to the true ML estimate, $s^\infty = 0$ if $(r^2 - \sigma_n^2) < 0$ or $\pm\sqrt{(r^2 - \sigma_n^2)/\sigma_a^2}$ otherwise. □

In both of the above examples the EM algorithm converged to the true ML estimate. This, however, is not always the case and may depend on the form of the likelihood function (see [1, 2]), and/or the quality of the initial estimate.

Next, we apply the EM algorithm to the random-phase and fading channels.

3 The Random Phase Channel

Let $\mathbf{s} = (s_1, s_2, \cdots, s_N)$ be the vector containing the complex modulation symbols, $\mathbf{D}$ be a diagonal matrix with the elements of $\mathbf{s}$ as diagonal elements, and $\mathbf{\Theta} = (e^{j\theta_1}, e^{j\theta_2}, \cdots, e^{j\theta_N})$ be the vector containing the random phase rotations in each symbol interval. Then the received (complex) vector $\mathbf{r}$ can be expressed as

$$\mathbf{r} = \mathbf{D}\mathbf{\Theta} + \mathbf{N}, \tag{5}$$

where $\mathbf{N}$ is a zero mean, i.i.d., complex, Gaussian noise vector. After suitable normalization and without loss of generality, we assume that for all k, $E(s_k) = 0$, $E(|s_k|^2) = 1$, and that the variance of each component of N_k is $\mathcal{N}_0/2E_s$ (E_s is the average energy in the transmitted signals).

The incomplete data vector is the vector $\mathbf{r}$ whose components are defined by (5), and the parameter vector to be estimated is the modulation sequence $\mathbf{s}$. Let the complete data $\mathbf{x}$ consist of the incomplete data along with knowledge of the random phase vector $\mathbf{\Theta}$ i.e., $\mathbf{x} = (\mathbf{r}, \mathbf{\Theta})$. For the constant phase-error case when $\theta_i = \theta$, the E-step of the algorithm yields

$$\begin{aligned} Q(\mathbf{s}|\mathbf{s}^i) &= \Re\left\{\mathbf{r}^\dagger \mathbf{s} e^{-j\psi(\mathbf{r},\mathbf{s}^i)}\right\} \\ &= \sum_k \Re\left\{r_k^* e^{-j\psi(\mathbf{r},\mathbf{s}^i)} s_k\right\}, \end{aligned} \tag{6}$$

where $\psi(\mathbf{r}, \mathbf{s}^i)$ is the angle $\mathbf{r}^\dagger\mathbf{s}$, assuming that sequence $\mathbf{s}$ was sent. Note that during the maximization of (6) at the i-th step of the iteration, $\psi(\mathbf{r}, \mathbf{s}^i)$ is a known *number*, computed at the previous step.

Another expression equivalent to (6), but one that does not require the computation of trigonometric functions is obtained by multiplying the right-hand side of (6) by the constant $|\mathbf{r}^\dagger\mathbf{s}^i|$ to yield:

$$Q(\mathbf{s}|\mathbf{s}^i) = \Re\left\{\mathbf{r}^\dagger\mathbf{s} \cdot \left(\mathbf{r}^\dagger\mathbf{s}^i\right)^*\right\}. \tag{7}$$

Again, note that $\mathbf{r}^\dagger\mathbf{s}^i$ is a known number already computed at the previous iteration.

We now turn to the maximization step of the EM algorithm, where we distinguish between coded and uncoded transmission. First we observe from (6), (7) that in the absence of coding, maximizing $Q(\mathbf{s}|\mathbf{s}^i)$ with respect to sequences $\mathbf{s}$ is equivalent to maximizing each individual term in the sum, i.e., making symbol-by-symbol decisions. Then, if $\mathbf{s}^{i+1}$ is the maximizing sequence, its components are given by

$$s_k^{i+1} = \arg \max_{s_k} \Re\left\{r_k^* e^{j\psi(\mathbf{r},\mathbf{s}^i)} s_k\right\}, \tag{8}$$

where we have used the expression for $Q(\mathbf{s}|\mathbf{s}^i)$ in (6) for illustration, even though in a practical implementation (7) may be more attractive.

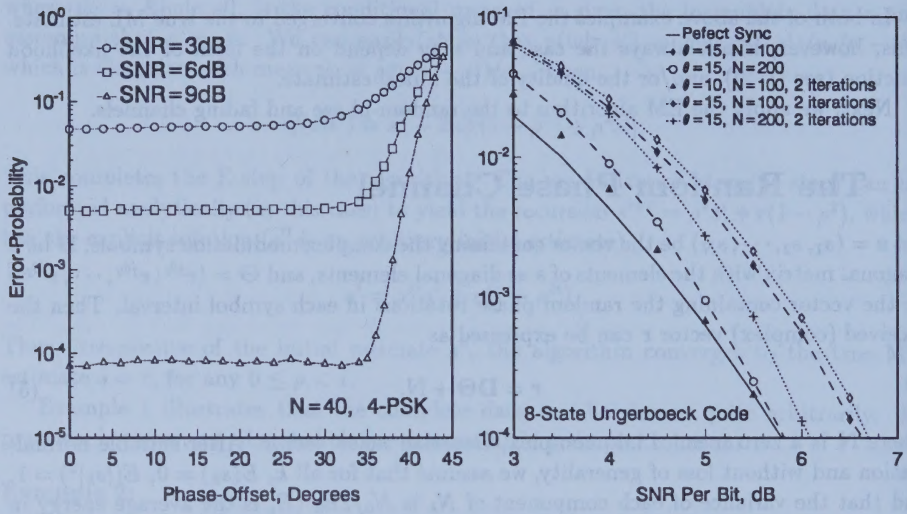


Figure 1: Error-Probability for the EM-based algorithm for a random-phase channel.

When trellis-coding is used, the maximization over all trellis sequences can be done efficiently using the Viterbi algorithm. It has been observed that the algorithm converges mostly in 2-3 iterations for practical SNR's, which may not be impractical for some low to medium rate applications. Even more importantly, very little performance is lost if the number of iterations is fixed to two, which has the advantage not only of a small but also *fixed* number of iterations.

At convergence, the EM algorithm provides also an estimate of the phase-offset besides the sequence estimate. In practice this estimate (or a filtered version of it) can be used as the initial estimate in the next observation window to further improve performance.

The first graph in Figure 1 shows performance as a function of phase offset for 4-PSK. For $N = 40$ the phase-offset threshold is about 35 degrees in this case, with practically constant performance below this value.

Table 1 shows the average number of iterations for convergence for 4-PSK. These results indicate that, on the average, convergence is mostly within two to three iterations for a range of practical SNR's.

The second graph in Figure 1 shows the symbol-error probability achieved by the 8-state Ungerboeck code as a function of the SNR per bit and initial phase-offsets of 10 and 15 degrees. Pessimistically, we assume that no phase-estimates in previous windows are utilized and all data blocks are processed independently. As can be seen from the figure, the algorithm achieves practically perfect performance for $\theta = 10$ degrees as the SNR is increased, even though the number of iterations was limited to two. For $\theta = 15$ degrees the algorithm looses about 0.8dB and is only slightly worse when the number of iterations is limited to two. This loss practically disappears as the block length increases.

Table 1: Average number of iterations for convergence as a function of the signal-to-noise ratio per bit and phase offset for 4-PSK with $N = 20$.

SNR, dB	Offset=10 degrees	Offset=20 degrees	Offset=30 degrees
4	2.40	3.06	3.77
6	2.15	2.63	3.26
8	2.03	2.31	2.91
10	2.00	2.07	2.66
12	2.00	2.00	2.32

4 The Fading Channel

We assume a fading channel with Rayleigh distributed amplitude fading and no phase coherence (uniform phase), which is equivalent to independent Gaussian amplitude fading on the in-phase and quadrature components of the transmitted symbols. If $\mathbf{D}$, $\mathbf{s}$ and $\mathbf{N}$ are as defined above, and $\mathbf{a}$ is a complex, zero mean, Gaussian vector with independent real and imaginary parts, then the received data $\mathbf{r}$ is given by

$$\mathbf{r} = \mathbf{D}\mathbf{a} + \mathbf{N}. \tag{9}$$

Much work has been done in designing receivers for this channel. Here, we apply the EM algorithm as another example of its use in an important communications channel.

Let the complete data $\mathbf{x}$ be the incomplete data $\mathbf{r}$ along with the random fading vector $\mathbf{a}$: $\mathbf{x} = (\mathbf{r}, \mathbf{a})$. Then the E-step of the algorithm yields (again skipping derivations)

$$Q(\mathbf{s}|\mathbf{s}^i) = \Re\left[\mathbf{r}^\dagger \mathbf{D}\hat{\mathbf{a}}_i\right] \tag{10}$$

where

$$\hat{\mathbf{a}}_i = E\left[\mathbf{a}|\mathbf{r}, \mathbf{s}^i\right] = \mathbf{Q}[\mathbf{Q} + \frac{\mathbf{I}}{SNR}]^{-1}(\mathbf{D}^i)^*\mathbf{r}. \tag{11}$$

A special case of the above model is when the fading process is a Markov process. In this case the required conditional moments can be generated at each step of the EM iteration using a Kalman filter.

When the fading is constant over a number of symbols, we can show that the expectation step of the algorithm becomes

$$Q(\mathbf{s}|\mathbf{s}^i) = \Re\left[\mathbf{r}^\dagger \mathbf{s}\hat{a}_i\right], \tag{12}$$

where

$$\hat{a}_i = \mathbf{r}\mathbf{s}^{i\dagger}. \tag{13}$$

As for the random phase case, the EM-based algorithm starts with an initial estimate of the fading vector, $\hat{a}_0$, and uses it in (10) (or (12)) to produce by maximization a sequence estimate. This sequence estimate is then used in (11) (or (13)) to produce the next fading estimate, and so on until convergence (within two to three iterations). At convergence, the algorithm produces both a sequence estimate and a fading vector estimate.

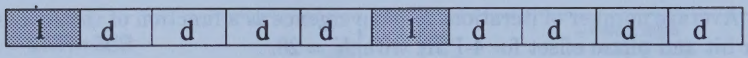


Figure 2: Illustration of received data for fading channel for $N = 5$, $L = 2$.

We first look at the case when the fading is constant over a number of symbols and consider first binary PSK. In this case the likelihood function cannot distinguish between a transmitted sequence $\mathbf{s}$ and its negative. Thus, it is necessary at the transmitter to limit the set of transmitted sequences of some length N to only half of the possible 2^N sequences, eliminating for every sequence $\mathbf{s}$ included its negative. This can be accomplished easily by fixing the first symbol in every transmitted sequence to be (for example) one, and then randomly modulating the remaining $(N-1)$ bits. This results in a code of rate $\rho = (N-1)/N$, which can be made as close to one as desired by increasing N. In making decisions, the receiver has available a vector of data $\mathbf{r}$ of length L codewords (or LN bits), for some integer L. Figure 2 illustrates the received data for $N = 5$ and $L = 2$.

In setting the initial fading estimate, the algorithm uses the fact that the symbols in positions 1, $(N+1)$, $\cdots, (L-1)N+1$ are known (all one in our case). An initial fading estimate is then easily obtained by summing the data in each of the L known bit positions and dividing by L:

$$\hat{a}_0 = \frac{1}{L} \sum_{k=0}^{L-1} r_{kN+1}.$$

Further, the maximization step of the algorithm need only be performed over the $L(N-1)$ modulated bits, and since these bits are randomly chosen, maximizing over the set of length $L(N-1)$ bit sequences is equivalent to making symbol-by-symbol decisions on each bit. In other words, if $\mathbf{s}^{i+1}$ is the sequence that maximizes (12), its components are easily obtained through $s_k^{i+1} = \operatorname{sgn}[\Re(r_k^* \hat{a}_i)]$, where $\operatorname{sgn}(\cdot)$ is the signum function. For the nonbinary case, the above symbol-by-symbol estimator is replaced by $s_k^{i+1} = \arg \max_{s_k} \Re(r_k^* s_k \hat{a}_i)$.

If coding is used, the maximization is over a constrained set which can be accomplished through use of the Viterbi algorithm for trellis-coded systems or in general through the use of whatever efficient decoding algorithm is available for the code.

The first graph in Figure 3 shows results of simulations for the ML receiver and the EM-based algorithm, limiting the number of iterations to two. The small value of N for the ML receiver was chosen to make implementing the ML receiver feasible, since in making sequence estimates this receiver must compute the likelihood-function for all 2^{N-1} sequences before it makes a decision. It is clear from the figure that the EM-based algorithm performs exactly as the ML algorithm for $N = 5$ and $L = 1$. As L is increased, the EM-based receiver makes better use of the channel memory and performs increasingly better, approaching closely the performance of a receiver that knows exactly what the fading is. Results for $N = 10$ and $L = 1, 2$ are also shown in the figure. The results for the ML and EM-based receivers are in terms of dB per *information* bit and account for the rate loss.

The second graph in Figure 3 presents simulation results for the performance of the EM-based algorithm on a non-static mobile communication channel for QPSK. The normalized doppler bandwidth parameter BT is 0.03 in this case, but similar results were obtained for other values. To simulate the fading process, the Cholesky factorization

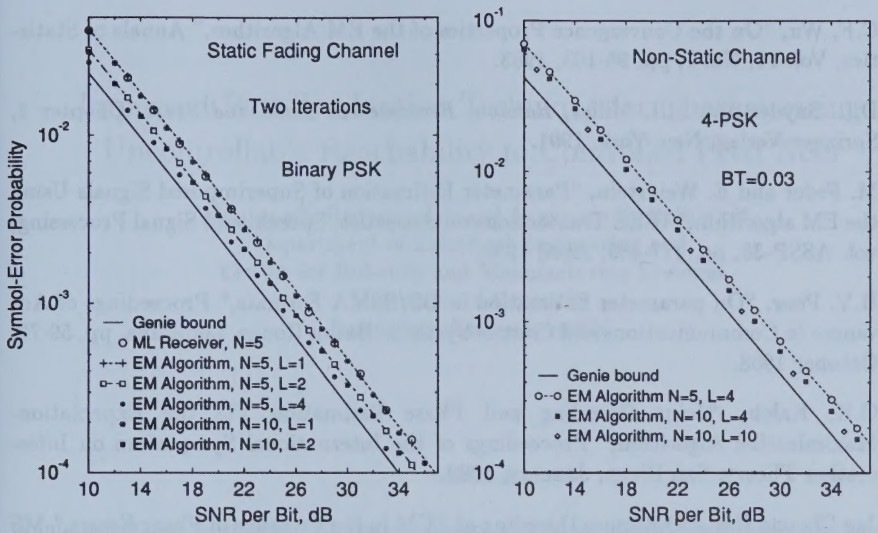


Figure 3: Error-probability of the EM-based algorithm for a fading channel.

was used to factor $\mathbf{Q}$ into the product of triangular matrices[2]. The lower-triangular matrix was then used to convert a white Gaussian sequence into one with the required covariance matrix with elements $Q_{ij} = J_0[2\pi BT(i-j)]$. The algorithm was initialized using fading estimates that were linearly interpolated from the data over the symbol intervals containing known symbols. As can be seen from the figure, the performance of the EM algorithm approaches closely that of the genie bound. The number of iterations for convergence was again fixed to two.

5 Conclusion

We have investigated the application of the EM algorithm to the problem of sequence estimation in the presence of random parameters, and specifically for random phase and fading channels. Simulation results were presented which show that the EM-based algorithms converge quickly to the ML sequence estimates, and can be used easily for both uncoded and trellis-coded systems. Perhaps the most important aspect of the EM-based algorithms for the applications considered is that they make it possible to produce maximum-likelihood estimates of long coded or uncoded sequences, a task that was previously too computationally demanding.

References

[1] A.P. Dempster, N.M. Laird and D.B. Rubin "Maximum-Likelihood from Incomplete Data Via the EM Algorithm," Journal, Royal Statistical Society, Vol. 39, pp. 1-17, 1977.

[2]For long fading vectors, factoring Q was numerically problematic. In this case, $(\mathbf{Q} + \mathbf{I}/SNR)$ was factored instead and used to generate the received faded data, since for PSK signaling $\mathbf{r}$ can be equivalently expressed as $\mathbf{r} = \mathbf{DW}$ where the covariance of $\mathbf{W}$ is $(\mathbf{Q} + \mathbf{I}/SNR)$.

[2] C.F. Wu, "On the Convergence Properties of the EM Algorithm," Annals of Statistics, Vol. 11, No. 1, pp. 95-103, 1983.

[3] D.L. Snyder and M.I. Miller, *Random Processes in Time and Space*, Chapter 3, Springer-Verlag, New York, 1991.

[4] M. Feder and E. Weinstein, "Parameter Estimation of Superimposed Signals Using the EM algorithm," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-36, pp. 477-489, April 1988.

[5] H.V. Poor, "On parameter Estimation in DS/SSMA Formats," Proceedings of Advances in Communications and Control Systems, Baton Rouge, Louisiana, pp. 59-70, October 1988.

[6] G.K. Kaleh, "Joint Decoding and Phase Estimation Via the Expectation-Maximization Algorithm," Proceedings of the International Symposium on Information Theory, San Diego, January, 1990.

[7] Jae Choong Han, "Optimum Decoding of TCM in the Presence of Phase Errors," MS Thesis, Electrical Engineering Department, Texas A&M University, August 1990.

[8] C.N. Georghiades and J.C. Han, "Optimum Decoding of Trellis-Coded Modulation in the Presence of Phase-Errors," Proceedings of the 1990 International Symposium and Its Applications (ISITA' 90), Hawaii, November 1990.

[9] C.N. Georghiades and D.L. Snyder, "The Expectation-Maximization Algorithm for Symbol Unsynchronized Sequence Detection," IEEE Transactions on Communications, vol. 39, pp. 54-61, January 1991.

[10] C.N. Georghiades, "Algorithms for Joint Synchronization and Detection," in *Coded Modulation and Bandwidth-Efficient Transmission*, Elsevier Science Publishers, 1992.

[11] S.M. Zabin and H.V. Poor, "Efficient Estimation of class A noise parameters via the EM algorithm," IEEE Transactions on Information Theory, vol. IT-37, pp. 60-72, January 1991.

[12] A. Ansari and R. Viswanathan, "Application of Expectation-Maximization Algorithm to the Detection of a Direct-Sequence Signal in Pulsed Noise Jamming," IEEE Transactions on Communications, Vol. 41, No. 8, pp. 1151-1154, August 1993.

[13] J.W. Modestino, "Reduced-complexity Iterative Maximum-Likelihood Sequence Estimation on Channels with Memory," Proceedings of the International Symposium on Information Theory, San Antonio, Texas, January 1993.

Improved Path Evaluation Technique for Characterizing Uncontrollable Reachability in Controlled Petri Nets

Manjunath Hiriyannaiah and Lawrence E. Holloway*
Department of Electrical Engineering and
Center for Robotics and Manufacturing Systems
University of Kentucky
Lexington, Kentucky 40506

Abstract

A fundamental issue in forbidden state avoidance control of discrete event systems is the identification of states which can uncontrollably lead to forbidden states. Recent work in [1, 2] has resulted in methods which characterize the uncontrollable reachability of forbidden states in controlled Petri nets (CtlPNs) through evaluation of expressions determined from the net structure. In this paper, we develop an improved method for evaluating such expressions that can reduce the overall computational effort of determining uncontrollable reachability from a given marking. Such overall savings, however, come at a cost of shifting some of the burden onto marking dependent operations. The method is illustrated with an example.

1 Introduction

In the state control of discrete event systems, the control problem is to ensure that the system never acheives a state outside a specified desired set, or equivalently, never achieves a state within a specified forbidden set. Due to the uncontrollable events in the system, determining a control is equivalent to evaluating the reachable set of states under the possible control choices [3, 4]. It is thus important to be able to characterize the set of uncontrollably reachable states.

In this paper we consider forbidden state avoidance in controlled Petri nets (CtlPNs) [5, 3]. The class of Petri net models we consider are capable of modeling the conflict represented by automata structures(e.g. [6]), as well as the process synchronization represented by marked graph structures (e.g. [7]). In [2], an algebraic characterization of PN reachability/coverability was introduced, and in [1], the method is discussed in a step-by-step manner. This paper builds directly upon the results in [8], which is a modification of [2]. In this paper, we introduce the 0 term into the algebra. By performing an initial evaluation of paths before manipulating the expression into the sum-of-products form necessary for evaluation, and then performing substitutions and simplifications with the 0 term, the computational effort of manipulating and evaluating an expression is reduced. We determine bounds for the number of product terms which must be evaluated in the full sum-of-products form versus the modified sum-of-products

*Please address all correspondence to L. E. Holloway at the above address. This work has been supported in part by NSF grant ECS-9308737, Rockwell International, NASA grant NGT-40049, and the Center for Robotics and Manufacturing Systems at the University of Kentucky

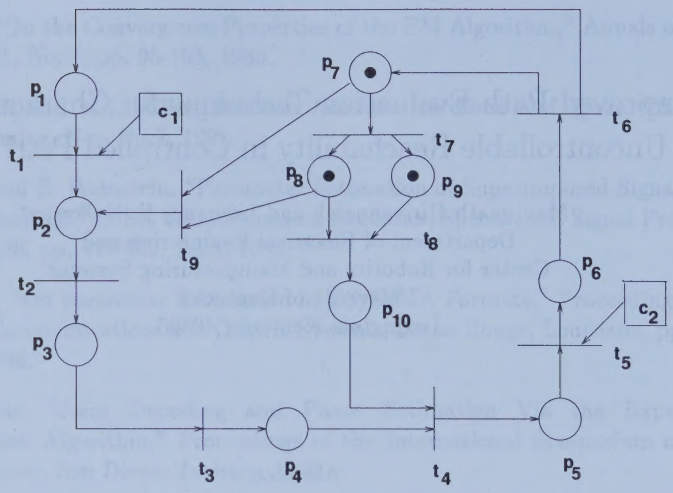


Figure 1: A controlled Petri net with $F = \{p_2, p_{10}\}$

form. These bounds give insight toward a complexity analysis of the methods. Details of the methods developed in this paper can be found in [9].

The following section reviews controlled Petri nets and forbidden state specifications. Section 3 describes the off-line evaluation of algebraic expressions, section 4 determines the on-line evaluation of expressions, section 5 describes the algorithm for on-line evaluation and the complexity of control method, section 6 describes the parameterization of Petri net model with an example and section 7 summarizes the paper.

2 Controlled Petri Nets

Controlled Petri nets have been defined in several papers, including [3, 7, 2], so only a brief description is given here. A controlled Petri net (CtlPN) is a tuple $\mathcal{G} = (\mathcal{P}, \mathcal{T}, \mathcal{E}, \mathcal{C}, \mathcal{B})$, where $\mathcal{P}$ is a finite set of *state places*, $\mathcal{T}$ is a finite set of *transitions*, $\mathcal{E} \subseteq (\mathcal{P} \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{P})$ is a set of directed arcs connecting state places and transitions, $\mathcal{C}$ is the finite set of control places, and $\mathcal{B} \subseteq (\mathcal{C} \times \mathcal{T})$ is the set of directed arcs associating control places with transitions. The notation ${}^{(p)}t$ and $t^{(p)}$ indicate the set of input and output state places, respectively, for the transition t. Similarly, ${}^{(t)}p$ and $p^{(t)}$ are respectively the input and output transitions for state place p.

A *path* $\pi \in (\{tp \mid t \in \mathcal{T},\ p \in \mathcal{P}\})^*$ is an alternating sequence of transitions and places representing a directed path in the CtlPN structure. Paths considered in this paper will always begin with a transition and end with a place. The starting transition for a path π is denoted t_π.

The *marking* (state) of a CtlPN is a function which associates a non-negative integer to each state place in the net. The integer represents the number of *tokens* in the place. The set of all markings is denoted $\mathcal{M}$. Figure 1 illustrates a controlled Petri net, where state places are indicated by circles, transitions by bars, and control places by squares. Tokens are represented by black dots in the state places.

A *control* $u : \mathcal{C} \longrightarrow \{0,1\}$ assigns a binary token count to each control place in the net. The set of all controls is denoted $\mathcal{U}$. Given two controls $u, u' \in \mathcal{U}$, control u' is said to be

more permissive than u if $u'(c) \geq u(c)$ for all $c \in \mathcal{C}$ and $u'(c) > u(c)$ for at least one $c \in \mathcal{C}$. A (nondeterministic) *feedback control law* is a function $\mathrm{U} : \mathcal{M} \longrightarrow 2^{\mathcal{U}}$ which assigns to each marking a set of possible controls. A set of transitions $T \subseteq \mathcal{T}$ is *state enabled* under a marking m when $m(p) \geq |p^{(t)} \cap T|$ for all $p \in^{(p)} t$. i. e., for each state place p, there exists at least as many tokens in p as there are output transitions to p in the set T. A set of transitions T is *enabled* under a marking m and a control u if it is both state enabled and $u(c) = 1$ for all $c \in^{(c)} t$ for all $t \in \mathcal{T}$. A state change occurs when a set of enabled transitions $T \subseteq \mathcal{T}$ *fires*, which results in a new marking m' defined by the following equation:

$$m'(p) = m(p) - |\, p^{(t)} \cap T \,| + |\, {}^{(t)}p \cap T \,|$$

Thus, the marking of each state place is decremented by the firing of an output transition of the state place and incremented by the firing of an input transition to the state place.

Given a marking m and a control u, a marking m' is said to be *immediately reachable* if there exists a set of transitions T such that T is enabled under m and u, and the firing of T results in the marking m'. The set of immediately reachable markings under a marking m and a control u is denoted $\Re_1(m, u)$. A marking m_k is said to be *reachable* from a marking m_0 under a control $u \in \mathcal{U}$ if there exists a sequence $(m_0 m_1 \ldots m_k)$ such that $m_i \in \Re_1(m_{i-1}, u)$ for $0 < i \leq k$. The set of all reachable markings from m under u is denoted $\Re_\infty(m, u)$.

The control goal considered for this work is expressed in terms of *forbidden conditions*. A forbidden condition is a set of places $F \subseteq \mathcal{P}$. The set of markings which satisfy a forbidden condition F is defined as M_F, where $m \in M_F$ if $m(p) \geq 1$ for all places $p \in F$. Given a set of forbidden conditions $\mathcal{F}$, the set $M_\mathcal{F}$ of forbidden markings for $\mathcal{F}$ is the union of M_F for all $F \in \mathcal{F}$. Our control goal is to prevent the system from reaching any forbidden marking $m \in M_\mathcal{F}$. We note that this is equivalent to a coverability problem, i.e. we must determine a control to ensure that no reachable marking contains this minimum number of tokens [10].

The methods we describe in this paper are based on characterizing the structure of the net. A path π is an alternating sequence of places and transitions representing a directed path in G. The set of all paths for the net G is denoted Π_G. We define the *uncontrolled subnet* for p as the subnet of G consisting of all transitions, arcs, and places for which there is an uncontrolled path leading to p.

Given a net $\mathcal{G}$, the set of *branching places* is defined as $D := \{p \in \mathcal{P} \mid |p^{(t)}| > 1\}$, and the set of *joining places* is defined as $J_F := \{p \in \mathcal{P} \mid |\,{}^{(t)}p| > 1\}$. Define $P_F = D \cup F \cup \{p \mid \exists t \in p^{(t)} \text{ s.t. } t^{(p)} \cap J_F \neq \emptyset\}$.

Definition 1 : Given a set of places $F \subseteq \mathcal{P}$ and some place $p \in P_F$, define the set of *precedence paths* $\Pi_F(p) \subseteq \Pi_\mathcal{G}$ for p as the set of all paths $(p_1 t_1 \ldots t_{n-1} p_n)$ such that

1. Path ends at p: $p_n = p$;
2. Path starts at joining place, or output of break transition, or cycleback: If ${}^{(t)}p_1 \neq \emptyset$, then either $|\,{}^{(t)}p_1| > 1$ or ${}^{(t)}p_1 \cap (T_c \cup \pi) \neq \emptyset$ or ${}^{(p)}({}^{(t)}p_1) \cap (D \cup F) \neq \emptyset$;
3. Path contains no cycles: $p_i \neq p_j$ and $t_i \neq t_j$ for $1 \leq i \neq j \leq n$;
4. Path contains no controlled transitions: $\pi \cap T_c = \emptyset$;
5. Restriction on branching, joining, and forbidden places: For $1 \leq i \leq n$, $p_i \in D \cup F$ implies $i = n$, and $p_i \in J_F$ implies $i = 1$.

The set of precedence paths for set F is denoted as $\Pi_F := \cup_{p \in P_F} \Pi_F(p)$. The conditions in the above definition are straightforward. The last condition indicates that only the ending place can be a departing place or in F, and only the beginning place can be joining place.

3 Path Expressions

In this section, we define an algebra for determining reachability of sets of forbidden states. This set is defined using paths in $\Pi_{\mathcal{G}}$ as defined in the previous section, and using strings of transitions $\sigma \in T^*$. The set of transition strings T^* includes the empty string ϵ. In this paper, we also introduce in the algebra the symbol 0, which will later be used in evaluating expressions.

Definition 2 : The set EXPR of expressions in the algebra is defined recursively using the three binary operators $+$, $\cdot$, and the transition string prefix operation:

0. $0 \in$ EXPR
1. $\pi \in \Pi_{\mathcal{G}} \Longrightarrow \pi \in$ EXPR
2. $\mathbf{expr}_1 \in$ EXPR, $\mathbf{expr}_2 \in$ EXPR $\Longrightarrow (\mathbf{expr}_1 + \mathbf{expr}_2) \in$ EXPR
3. $\mathbf{expr}_1 \in$ EXPR, $\mathbf{expr}_2 \in$ EXPR $\Longrightarrow (\mathbf{expr}_1 \cdot \mathbf{expr}_2) \in$ EXPR
4. $\mathbf{expr} \in$ EXPR and $\sigma \in T^* \Longrightarrow {}^{\sigma}(\mathbf{expr}) \in$ EXPR.

The operation between transition strings and expressions, given in item 4 above, is referred to as a transition string prefix operation.

Given any two distinct sequences $\sigma_1, \sigma_2 \in T^*$, σ_1 and σ_2 are said to be in conflict if $\sigma_1 = \sigma_1' t_1 \sigma$ (i.e. the concatenation of strings σ', t_1, and σ) and $\sigma_2 = \sigma_2' t_2 \sigma$ for some common suffix σ, where t_1 and t_2 are in conflict.

Definition 3 : Given a product form expression *prod* and some $\pi \in \Pi(prod)$, the *token threshold* $Q(prod, \pi)$ is the maximum number of strings in $\Sigma(prod, \pi)$ that are jointly in conflict.

A *marking valuation predicate* is function that assigns a binary value to each $\mathbf{expr} \in$ EXPR. In this paper, we focus on the predicate Λ_m, which indicates whether a set of markings represented by an expression is uncontrollably reachable. In [3, 8], an additional predicate, Δ_m, is introduced and it is shown how the two predicates can be used to determine controls to prevent a set of markings from being uncontrollably reached. In this paper, we focus on the predicate Λ_m, defined as follows:

Definition 4 : Given a marking m for the net G, the *uncontrollability predicate* for m is the binary valuation Λ_m : EXPR $\longrightarrow \{0, 1\}$ that satisfies the following properties:

0. $\Lambda_m(0) = 0$
1. Λ_m satisfies the following axioms, for any expressions $e_1, e_2, e_3 \in$ EXPR:

1. Sum is Disjunctive:	$\Lambda_m(e_1 + e_2) = 1$ iff $\Lambda_m(e_1) = 1$ or $\Lambda_m(e_2) = 1$
2. Distributive $\cdot$ over $+$:	$\Lambda_m(e_1(e_2 + e_3)) = \Lambda_m(e_1e_2 + e_1e_3)$
3. Distributive $\diamond$ over $+$:	$\Lambda_m({}^{\sigma}(e_2 + e_3)) = \Lambda_m({}^{\sigma}e_2 + {}^{\sigma}e_3)$
4. Distributive $\diamond$ over $\cdot$:	$\Lambda_m({}^{\sigma}(e_1e_2)) = \Lambda_m({}^{\sigma}(e_1)\,{}^{\sigma}(e_2))$
5. Concatenation through $\diamond$:	If $\sigma = \sigma_1\sigma_2$, then $\Lambda_m({}^{\sigma_1}({}^{\sigma_2}e_1)) = \Lambda_m({}^{\sigma}e_1)$

2. For any product *prod* in EXPR, $\Lambda_m(prod) = 1$ iff *prod* does not contain term 0 and for all $\pi \in \Pi(prod)$,

$$m(\pi) \geq Q(prod, \pi)$$

Note that item 2 in the definition of Λ_m is in terms of product terms, and it is typically necessary to convert an expression into a sum of product form to be able to evaluate it. Such a sum-of-products form can always be achieved using the axioms listed in item 1 of the above definition [8].

The expressions we consider are constructed by recursively performing an expansion operation on elementary expressions. These are defined below.

Definition 5 : Given a set of places $F \subseteq \mathcal{P}$ and a place $p \in P_F$, let $\{\pi_1, \pi_2, \ldots \pi_n\} = \Pi_F(p)$. Define $\mathtt{basic}_F(p) \in \text{EXPR}$ as the product

$$\mathtt{basic}_F(p) := \pi_1 \cdot \pi_2 \cdots \pi_n$$

Definition 6 : Given a CtlPN G and a set of places $F \subseteq \mathcal{P}$, the $\mathtt{expand}_F$ operation is a mapping $\mathtt{expand}_F : \text{EXPR} \rightarrow \text{EXPR}$ defined as follows:

1. For any $\pi \in \Pi_G$,

$$\mathtt{expand}_F(\pi) := \pi + \sum_{t \in T_\pi - T_c} {}^{t}\left(\prod_{p \in {}^{(p)}t \cap (D_F \cup F)} \mathtt{basic}_F(p) \right)$$

2. For any expression $e \in \text{EXPR}$, $\mathtt{expand}_F(e)$ is found by replacing in e (nonrecursively) each occurrence of $\pi \in \Pi_F(p)$ with $\mathtt{expand}_F(\pi)$.

In [8], an expression $\mathtt{expr}_F(p)$ is defined as the fixpoint of a recursive expansion of $\mathtt{basic}_F(p)$, such that for any marking m, $\Lambda_m(\mathtt{basic}_F(p)) \leq \Lambda_m(\mathtt{expr}_F(p))$ and $\Lambda_m(\mathtt{expr}_F(p)) = \Lambda_m(\mathtt{expand}_F(\mathtt{expr}_F(p)))$. Furthermore, it is shown that for any net and any F and $p \in F$, there exists such an $\mathtt{expr}_F(p)$ with a finite number of terms. In this paper, we exclusively consider nets without uncontrolled cycles, which allows a very direct construction of $\mathtt{expr}_F(p)$. In constructing an expression given such a net, whenever a term π is expanded, we associate a tag to the term π in $\mathtt{expand}_F(\pi)$ to indicate its prior expansion. The construction then proceeds as follows:

1 $\mathtt{expr}_F(p) \Leftarrow \mathtt{basic}_F(p)$

2 Replace each untagged term π in $\mathtt{expr}_F(p)$ with $\mathtt{expand}_F(\pi)$. Let the resulting expression by $\mathtt{expr}_F(p)$.

3 While there exists some untagged term in $\mathtt{expr}_F(p)$, repeat step 2.

Under the following condition, the expression $\mathtt{expr}_F(p)$ with the predicate Λ_m characterizes whether there exists an uncontrollably reachable marking for which p is marked.

Definition 7 : A CtlPN G satisfies condition C1 for forbidden condition F if for any $p_1, p_2 \in P_F$, if there exists a $t_1 \in p_1^{(t)} - p_2^{(t)}$ and $t_2 \in p_2^{(t)} - p_1^{(t)}$ such that ${}^{(p)}t_1 \cap {}^{(p)}t_2 \neq \emptyset$, then $T_{\pi_1} \cap T_{\pi_2} = \emptyset$ for all $\pi_1 \in \Pi_F(p_1)$, $\pi_2 \in \Pi_F(p_2)$.

The following lemma then identifies the use of expression evaluation to characterize uncontrollable reachability.

Lemma 1 [8] Given a CtlPN G satisfying condition C1 for forbidden set F, for a place $p \in F$ and a given marking m, there exists an uncontrollably reachable marking $m' \in \mathcal{R}_\infty(m, u_{zero})$ such that $m'(p) \geq 1$ if and only if $\Lambda_m(\mathtt{expr}_F(p)) = 1$.

It is shown in [8] that determining the reachability of forbidden markings corresponding to a forbidden condition F is equivalent to determining the reachability of a place marking in an augmented net. To avoid the introduction of additional notation, without loss of generality we will restrict our discussion to forbidden place markings.

4 On-line expression manipulation

From the definition of the Λ_m valuation, an expression must be manipulated into a sum-of-products form to determine whether the forbidden markings that the expression represents are uncontrollably reachable. Such a manipulation can be done off-line since it is marking-independent, allowing the resulting products to each be quickly evaluated on-line for a given marking. However, for some net structures, the sum-of-products form may contain a large number of product terms, and be considerably less compact than the initial expression. An alternative approach is to manipulate the expression into sum-of-products form on-line for a given marking, but use our knowledge of the marking to limit the amount of manipulation needed. In particular, we perform an initial evaluation to determine the paths which will necessarily evaluate to zero. By eliminating these terms before manipulation, the total number of steps and the number of terms in the final expression are reduced. Such an on-line method may be very effective if it is typical that many of the precedence paths for p will be unmarked.

For a given marking m, we create this reduced expression as follows:

1 $N(\mathbf{expr}_F(p), m) \Leftarrow \mathbf{expr}_F(p)$
2 For each path π in $N(expr_F(p), m)$ with $m(\pi) = 0$, replace π with term 0.
3 Recursively perform the following substitutions for any expression $\mathbf{e}$ appearing within $\mathbf{expr}_F(p)$:
$\mathbf{e} + 0 \Leftarrow \mathbf{e}$ and $0 + \mathbf{e} \Leftarrow \mathbf{e}$
$\mathbf{e} \cdot 0 \Leftarrow 0$ and $0 \cdot \mathbf{e} \Leftarrow 0$.

The expression $N(\mathbf{expr}_F(p))$ can then be manipulated into a sum-of-products form in the same manner that $\mathbf{expr}_F(p)$ is. The definition of $N(\mathbf{expr}_F(p), m)$ leads to the following lemma.

Lemma 2 Given a CtlPN G, forbidden condition F, and place $p \in F$, for any marking m,

$$\Lambda_m(\mathbf{expr}_F(p)) = \Lambda_m(N(\mathbf{expr}_F(p), m))$$

Proof: For any $\pi \in \Pi_G$ with $m(\pi) = 0$, for any product $\mathbf{expr} \cdot \pi$, $Q((\mathbf{expr} \cdot \pi), \pi) \geq 1$, so $\Lambda_m(\mathbf{expr} \cdot \pi) = 0 = \Lambda_m(\mathbf{expr} \cdot 0) = \Lambda_m(0)$. Similarly, from axiom 1 in definition 4, we have $\Lambda_m(\mathbf{expr} + \pi) = \Lambda_m(\mathbf{expr}) + \Lambda_m(\pi) = \Lambda_m(\mathbf{expr})$, and $\Lambda_m(\mathbf{expr} + 0) = \Lambda_m(\mathbf{expr})$. Thus, replacing unmarked paths by zero and recursively performing the substitutions on sums and products with 0 does not affect the final valuation. □

The above lemma and lemma 1 lead directly to the following result:

Lemma 3 Given a CtlPN G which satisfies condition C1 for forbidden set F, for a given marking m and place $p \in F$, there exists an uncontrollably reachable marking $m' \in \mathcal{R}_\infty(m, u_{zero})$ such that $m'(p) \geq 1$ if and only if $\Lambda_m(N(\mathbf{expr}_F(p)), m) = 1$.

Since evaluation of an expression using Λ_m requires evaluation of each product in the sum-of-products form of the expression, it is useful to determine a bound on the number of products that will exist. Such a bound can serve as a measure of the computational effort to determine the sum-of-products form, and as a measure of the computational effort to evaluate Λ_m for the sum-of-products form. We will determine such a bound for a given marking for both $\mathbf{expr}_F(p)$

and $N(\mathtt{expr}_F(p), m)$, and use it to compare the two expressions for evaluating uncontrollable reachability.

The number of product terms in an expression $\mathtt{expr}_F(p)$ is highly dependent upon the net structure. To determine a bound, we parameterize the uncontrollable subnet for p using four parameters. Define the following for the uncontrolled subnet for place p:

n_1 = Maximum cardinality of $T_\pi - T_c$ for all $\pi \in \Pi_F$

n_2 = Maximum cardinality of $\Pi_F(p')$ over all p' in P_F

n_3 = Maximum cardinality of ${}^{(p)}t \cap P_F$ for any $t \in T_\pi - T_c$ and path $\pi \in \Pi_F$.

n_4 = Maximum number of places in P_F between place $p \in F$ and a controlled transition in the uncontrolled subnet.

The following theorem provides an upper bound on the number of product terms in the sum-of-products form of $\mathtt{expr}_F(p)$.

Theorem 1 Given a net G and forbidden set F, for any $p \in F$, let n_1, n_2, n_3 and n_4 be the parameters characterizing the uncontrolled subgraph of p. The upper bound on the number of product terms in the sum-of-products form of $\mathtt{expr}(p)$ for a given place $p \in F$ is K_{n_1,n_2,n_3,n_4}, determined by the recursive equation

$$K_{n_1,n_2,n_3,i} = \begin{cases} 1 & \text{for } i = 0 \\ (1 + n_1(K_{n_1,n_2,n_3,i-1})^{n_3})^{n_2} & \text{for } i > 0 \end{cases}$$

Proof:

We prove the theorem by induction on n_4.

Initial step: First we consider the case $n_4 = 0$. From definition 5, we know that $\mathtt{expr}_F(p)$ contains one product term. For $n_4 = 0$, there are no expansions, so $\mathtt{expr}_F(p) = \mathtt{basic}_F(p)$, and $K_{n_1,n_2,n_3,0} = n_1$.

Now, consider the case $n_4 > 0$, where we assume that K_{n_1,n_2,n_3,n_4-1} is known.

Induction step: By its definition, $\mathtt{expr}_F(p)$ is the recursive expansion of $\mathtt{basic}_F(p)$, which is the product of paths in $\Pi_F(p)$. From the definition of $\mathtt{expand}_F(\pi)$, for each such path π, its recursive expansion includes products of the recursive expansion of terms $\mathtt{basic}_F(p')$ for $p' \in {}^{(p)}t \cap P_F$, for $t \in T_\pi - T_c$. Since p' is between p and some controlled transition, and there were at most n_4 places in P_F between p and any controlled transition in the uncontrolled subnet, then there are at most $n_4 - 1$ such places for p'. The recursive expansion of $\mathtt{basic}_F(p')$ thus gives K_{n_1,n_2,n_3,n_4-1} terms. There are at most n_3 places in ${}^{(p)}t \cap P_F$ for any $t \in T_\pi - T_c$, so for each such t there are $(K_{n_1,n_2,n_3,n_4-1})^{n_3}$ products. Since $\mathtt{expand}_F(\pi)$ includes the term π, and since there are at most n_1 transitions $t \in T_\pi - T_c$, then there will be $(1 + n_1(K_{n_1,n_2,n_3,n_4-1})^{n_3})$ products in the recursive expansion of π. The expression $\mathtt{basic}_F(p)$ is the product of at most n_2 paths, which leads to a total of $(1 + n_1(K_{n_1,n_2,n_3,n_4-1})^{n_3})^{n_2}$ products in $\mathtt{expr}_F(p)$, which is the recursive expansion of $\mathtt{basic}_F(p)$. □

The number of product terms in the sum-of-products form of $N(\mathtt{expr}_F(p), m)$ is marking dependent, so we resort to determining a bound on the expected value of the number of terms. Let M be a set of markings such that for every path $\pi \in \Pi_F$, q is the probability that $m(\pi) > 0$ for randomly selected $m \in M$. Furthermore, we assume that the set sample set M is such that for any two different paths π, π', the probability that path π is marked is independent of the probability that path π' is marked. This second assumption on the independence of paths will not be true for most marking sets M, since paths will sometimes overlap. However, we choose our sample set of markings M with such a constraint to facilitate determination of a rough estimate on the number of product terms that are expected to occur in $N(\mathtt{expr}_F(p), m)$.

Theorem 2 Consider a net G and forbidden set F, such that for some n_1, n_2, and n_3, $|T_\pi - T_c| = n_1$ for all $\pi \in \Pi_F$, $|\Pi_F(p)| = n_2$ for all $p \in P_F$, and $|{}^{(p)}t \cap P_F| = n_3$ for all $t \in T_\pi - T_c$, $\pi \in \Pi_F$. Let M be a set of markings such that for any $\pi, \pi' \in \Pi_F$, $\text{Prob}[m(\pi) > 0] = \text{Prob}[m(\pi') > 0] = q$, and the probability of $m(\pi) > 0$ is independent of the probability of $m(\pi') > 0$. For any place $p \in P_F$ such that there are n_4 places in P_F between any controlled transition and p in the uncontrolled subgraph of p, the expected value of the number of product terms (which are not term 0) in the sum-of-products form of $N(\mathbf{expr}_F(p), m)$ is $\tilde{K}_{n_1,n_2,n_3,n_4}(q)$, determined by the recursive equation

$$\tilde{K}_{n_1,n_2,n_3,i}(q) = \begin{cases} q^{n_2} & \text{for } i = 0 \\ (q + n_1(\tilde{K}_{n_1,n_2,n_3,i-1}(q))^{n_3})^{n_2} & \text{for } i > 0 \end{cases}$$

Proof: We prove the theorem by induction on n_4.

Initial step: First we consider the case $n_4 = 0$, so $\mathbf{expr}_F(p) = \mathbf{basic}_F(p)$. From definition 5, $\mathbf{basic}_F(p)$ is the product of n_2 paths. In the construction of $N(expr_F(p), m)$, any path which was unmarked was replaced by 0. The probability that a path was not removed is the probability that it was marked, which is q. The probability that none of the n_2 paths in $\mathbf{basic}_F(p)$ expression was replaced by zero is thus q^{n_2}, which is then the probability that $\mathbf{basic}_F(p)$ was not replaced by zero in the substitutions done in creating $N(\mathbf{expr}_F(p), m)$. The expected value of the number of product terms (that are not the term 0) in $N(\mathbf{expr}_F(p), m)$ is q^{n_2}.

Induction step: For any $\pi \in \Pi_F(p)$ and any $t \in T_\pi - T_c$, there is a product of n_3 expressions $\mathbf{basic}_F(p')$ in $\mathbf{expand}_F(\pi)$, where $p' \in {}^{(p)}t \cap P_F$. For each such p', from the induction, we have $\tilde{K}_{n_1,n_2,n_3,n_4-1}$ as the expected number of product terms in the recursive expansion of $\mathbf{basic}_F(p')$. Consequently, the expected value of the number of product terms resulting from the product of n_3 of these expressions is $(\tilde{K}_{n_1,n_2,n_3,n_4-1})^{n_3}$. Since there are n_1 transitions $t \in T_\pi - T_c$, then each such $\mathbf{expand}_F(\pi)$ contains n_1 of these expressions. Since path π has probability of being marked as q (and thus probability of contributing another product as q), then the expected value for each $\mathbf{expand}_F(\pi)$, $\pi \in \Pi_F(p)$, is $(q + n_1(\tilde{K}_{n_1,n_2,n_3,n_4-1})^{n_3})$. There are n_2 paths in $\Pi_F(p)$, so the expected number of products is $(q + n_1(\tilde{K}_{n_1,n_2,n_3,n_4-1})^{n_3})^{n_2}$. □

5 Example

To illustrate the results of the previous section, consider the uncontrolled subnet for p_1 as shown in figure 2, where $P_F = \{p_1, p_4, p_5, p_6, p_7\}$. The set of precedence paths Π_F consists of the following:

The precedence paths are as follows:

π_1	$= (p_2 t_1 p_1)$	π_2	$= (p_3 t_1 p_1)$
π_3	$= (p_8 t_6 p_4)$	π_4	$= (p_9 t_6 p_4)$
π_5	$= (p_{10} t_7 p_5)$	π_6	$= (p_{11} t_7 p_5)$
π_7	$= (p_{12} t_8 p_6)$	π_8	$= (p_{13} t_8 p_6)$
π_9	$= (p_{14} t_9 p_7)$	π_{10}	$= (p_{15} t_9 p_7)$

The basic path expressions are:

$\mathbf{basic}_F(p_1) = \pi_1 \pi_2$ $\quad$ $\mathbf{basic}_F(p_4) = \pi_3 \pi_4$

$\mathbf{basic}_F(p_5) = \pi_5 \pi_6$ $\quad$ $\mathbf{basic}_F(p_6) = \pi_7 \pi_8$

$\mathbf{basic}_F(p_7) = \pi_9 \pi_1 0$

The full expression for p_1 is then determined to be :

$$\mathbf{expr}_F(p_1) = (\pi_1 + {}^{t_2}(\pi_3 \pi_4) + {}^{t_3}(\pi_5 \pi_6))(\pi_2 + {}^{t_4}(\pi_7 \pi_8) + {}^{t_5}(\pi_9 \pi_{10}))$$

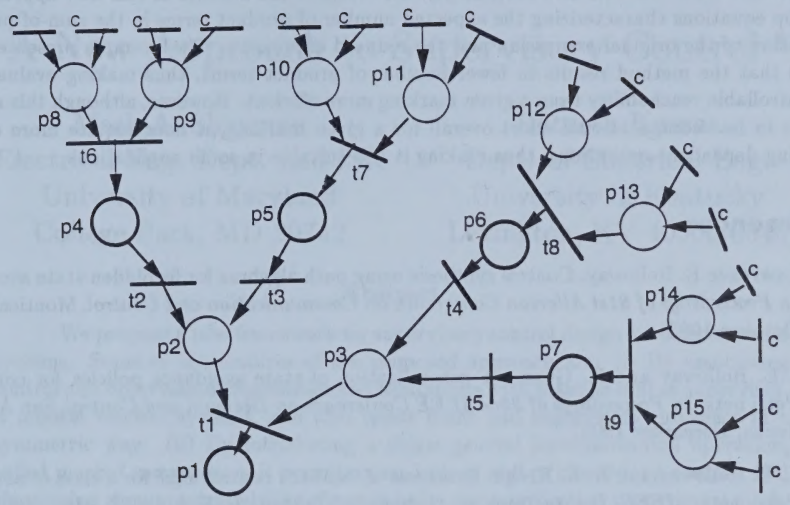


Figure 2: An example uncontrolled subnet for p_1 with $\{p_1, p_4, p_5, p_6, p_7\} = P_F$ and parameters $n_1 = 2$, $n_2 = 2$, $n_3 = 1$, $n_4 = 1$.

For the net shown in figure 2, the parameters are: $n_1 = 2, n_2 = 1, n_3 = 2$, and $n_4 = 1$. For these parameters, the number of product terms in $\mathbf{expr}_F(p_1)$ is given by $K_{(n_1,n_2,n_3,n_4)} = 9$. If each path has a $q = 0.5$ probability of being marked, then the expected value for the number of product terms in $N(\mathbf{expr}_F(p_1), m)$ is determined to be $\tilde{K}_{(n_1,n_2,n_3,n_4)}(0.5) = 1$.

It is clear from the above example that performing a preliminary substitution to determine $N(\mathbf{expr}_F(p), m)$ will often significantly reduce the number of product terms that must eventually be evaluated. Although each product term can be very easily and quickly evaluated individually, it is also apparent from the results of theorem 1 that some net structures can lead to a large number of product terms, even from a relatively compact initial expression. The reduction in total expected product terms can be a significant savings in some cases, although it moves the equation manipulation from an off-line operation to an on-line, marking dependent operation.

6 Summary

In this paper we have presented an alternative method for evaluating uncontrollably reachability of states represented by forbidden conditions. An algebraic expression characterizing the uncontrollable reachability of a set of states is determined using the method of [8]. To evaluate the expression requires manipulation of the expression to determine a sum-of-products form. For some nets, this sum-of-products form may be considerably less compact than the initial expression. In this paper, we introduce a term 0 into the algebra. By substituting the term for unmarked paths in the original expression, determining the sum-of-products form requires considerably less manipulation, and the resulting expression is typically much more compact than

the sum-of-products form of the original expression. This implies there are fewer terms to evaluate to determine uncontrolled reachability. To compare the benefits of this new approach, we develop equations characterizing the expected number of product terms in the sum-of-products expansion of the original expression and the reduced expression. The example presented illustrates that the method results in fewer number of product terms, thus making evaluation of uncontrollable reachability from a given marking more efficient. However, although this method results in less computational effort overall for a given marking, it does require more on-line, marking-dependent operations, thus making it less suitable in some applications.

References

[1] Lawrence E. Holloway. Control synthesis using path algebras for forbidden state avoidance. In *Proceedings of 31st Allerton Conference on Communication and Control*, Monticello, IL., October 1993.

[2] L.E. Holloway and X. Guan. A generalization of state avoidance policies for controlled Petri nets. In *Proceedings of 32nd IEEE Conference on Decision and Control*, San Antonio, Texas, December 1993.

[3] L. E. Holloway and B. H. Krogh. Synthesis of feedback control logic for a class of controlled Petri nets. *IEEE Transactions on Automatic Control*, 35(5):514–523, May 1990. Also appears in *Discrete Event Dynamic Systems: Analyzing Complexity and Performance in the Modern World*, edited by Y.C. Ho, IEEE Press, New York, 1992.

[4] Y. Li and W.M. Wonham. Control of vector discrete-event systems II – controller synthesis. *IEEE Transactions on Automatic Control*, 39(3), March 1994. to appear.

[5] B. H. Krogh. Controlled Petri nets and maximally permissive feedback logic. *Proceedings of 25th Annual Allerton Conference*, Sept. 1987. University of Illinois, Urbana.

[6] R.K. Boel, L. Ben-Naom, and V. Van Breusegem. On forbidden state problems for closed colored Petri nets. In *Proceedings of 12th IFAC World Congress*, Sidney, Australia, July 1993.

[7] Bruce H. Krogh and Lawrence E. Holloway. Synthesis of feedback control logic for discrete manufacturing systems. *Automatica*, July 1991.

[8] L.E. Holloway, X. Guan, and Lan Zhang. A generalization of state avoidance policies for controlled Petri nets. Technical Report CRMS-TR-93-2, Center for Robotics and Manufacturing Systems, University of Kentucky, Lexington, Kentucky, 1993. submitted to IEEE Transactions on Automatic Control. An earlier version of this paper appeared in *Proceedings of 32nd IEEE Conference on Decision and Control*, San Antonio, December 1993.

[9] Manjunath Hiriyannaiah. Improved path evaluation technique for control of an automated manufacturing system. Master's thesis, University of Kentucky, Lexington, Kentucky, May 1994.

[10] L. E. Holloway and B. H. Krogh. Controlled Petri nets: a tutorial survey. In *Proceedings of 11th International Conference on Analysis and Optimization of Systems: Discrete Event Systems*, Sophia-Antipolis, France, June 1994.

A New Approach to Supervisory Control *

Mark A. Shayman
Electrical Eng. Dept. and ISR
University of Maryland
College Park, MD 20742

Ratnesh Kumar
Dept. of Electrical Eng.
University of Kentucky
Lexington, KY 40506-0046

Abstract

We propose a new framework for supervisory control design for discrete event systems. Some of the features of the proposed approach are: (i) By associating control and observation capabilities and limitations with plant as well as supervisor, it models *reactive* systems, and also treats plant and supervisory processes in a symmetric way. (ii) By introducing a single general interconnection operation, called *masked composition*, it permits open-loop as well as closed-loop control. (iii) By viewing the uncontrollability of events as having a projection type control mask, and considering more general non-projection type control as well as observation masks, it treats the controllability and observability of events in a unified way. (iv) It applies to both deterministic and nondeterministic plant models and supervisory design. The sublanguages of a given language that are realizable under control are closed under union. Hence, the supremal realizable sublanguage always exists. (v) By encapsulating control and observation masks with process logic to form *process objects*, and using a single type of interconnection operator to build complex process objects out of simpler component process objects, it provides a foundation for an *object oriented* approach to discrete event control.

1 Introduction

The standard supervisory control framework is ill-suited for modeling *reactive systems*. In traditional supervisory control [10, 7], the plant generates all events. The plant is a "closed system"; the only intervention that is possible is from the supervisor disabling some controllable events. In models with "command events" [1] or "driven events" [3], the supervisor can initiate some events, but the controlled system of the plant and supervisor is again a closed system. In a reactive system, the controlled plant is not a closed system. It interacts with an environment that can initiate events that may be observed by the plant, and whose occurrence may or may not require the participation of the plant. For example, the radar screen of a fighter aircraft may indicate the presence of another aircraft, but may not be able to identify it as friend or foe. Thus, the *plant*–as opposed to the supervisor–observes an event initiated by the environment. Furthermore, it has partial observation since it cannot distinguish between the two events corresponding to friend and foe. To model reactive systems, not only the supervisor, but the plant as

*This research was supported in part by the Center for Robotics and Manufacturing, University of Kentucky, in part by by the National Science Foundation under the Engineering Research Centers Program Grant CDR-8803012 and Grants ECS-9312587 and ECS-9409712, and the Minta Martin Fund for Aeronautical Research, University of Maryland.

well, must have control and observation capabilities and limitations. This is one of the features of the proposed new approach.

The standard supervisory control framework of Ramadge-Wonham [10] permits the control input from the supervisor to be changed only when an event is observed. In other words, non-constant open-loop control is not permitted. This is true even of the recent work of Inan [5] in which nondeterministic supervision is allowed. Requiring the control input to be constant between observed events is analogous to requiring that zero-order hold be used in sampled data systems. However, it is more realistic in that setting since the sampling rate can be chosen to be high. In general, the frequency of observable events is not a design parameter. Consider the problem of allocating a shared communication channel between two users under the assumption that the controller cannot observe the messages submitted by the users. Suppose that the controller can disable the access of either user, and the specification is that there be no interleaving of the messages from the two users. Since there are no observable events, the Ramadge-Wonham framework would require that the control input be forever constant. Consequently, one user would be permanently denied access to the channel. In contrast, by introducing the interconnection operation called *masked composition*, our framework permits a "silent" transition in the supervisor that allows the access to be changed from one process to the other. As such, masked composition generalizes many of the interconnection operators including strict synchronous composition [4], prioritized synchronous composition [3], etc., while maintaining the desired property of associativity.

In nearly all the work on supervisory control, the event set is partitioned into a set of controllable events and a set of uncontrollable events. The supervisor can disable any subset of controllable events. This can be equivalently viewed as having a projection-type *control mask*. While observation has been studied using non-projection masks, the same has apparently not been done for control. Yet it is entirely natural to have groups of events which must be simultaneously disabled or not disabled at all. For example, a machine may perform two operations and be subject only to on-off control. Thus, to disable one operation requires shutting the machine off, thereby disabling the other operation. Our framework includes arbitrary control masks, as well as arbitrary observation masks.

In [11, 8, 9], necessary and sufficient conditions for the solution to the supervisory control problem for nondeterministic plants (with ϵ-moves) are obtained under the restrictions that (1) the control and observation masks are natural projections and (2) either every controllable event is observable [11] or only deterministic supervisors are permitted [9]. Using the new framework, we are able to solve the supervisory control problem for nondeterministic plants without either of these restrictions. Moreover, when the supervisor is not restricted to be deterministic, then the sublanguages of a given language that are realizable as closed-loop behavior are closed under union; consequently, the supremal realizable sublanguage always exists.

It is highly desirable to develop an object-oriented approach to supervisory control of discrete event systems. Efforts to develop such an approach for continuous variable control systems have already begun [6]. The basic goal is to develop the capacity to build standard modules that could be taken off the shelf and interconnected in different ways to control many different plants to meet different possible specifications. In our approach, each process object consists of a logic submodule encapsulated with an input interface (observation mask) and output interface (control mask). The interfaces may be modified to reflect new sensor or actuator capabilities *without requiring any modification in the logic submodule.* When objects are interconnected by *masked composition* to

build complex systems or to impose control, there are no compatibility conditions (such as supervisor completeness) that require off-the-shelf objects to be modified as in the framework proposed by Fabian-Lennartson [2]. In addition, the associativity of masked composition makes our approach suitable for modular control.

An expanded version of this paper is available from the authors [12].

2 Process Object Representation

We use standard notation for nondeterministic state machines with ϵ-moves (NSM's). We use $L(P)$ and $L_m(P)$ to denote the generated and recognized languages of the NSM P. Σ denotes the set of events, and $\overline{\Sigma} := \Sigma \cup \{\epsilon\}$. $\Sigma_P(x)$ denotes the events defined in state x of P, and $\overline{\Sigma}_P(x) := \Sigma_P(x) \cup \{\epsilon\}$. δ_P is the transition function of P.

To be able to define masked composition in such a way as to be associative, it is necessary to distinguish between two types of transitions in an NSM P. The *real* transitions are transitions of the usual type; they can generate events, permit another process to generate events, and synchronize with events generated by another process. In contrast, *virtual* transitions can only give permission to and track events generated by another process. They cannot be used to generate (initiate) events on their own. (See Example 2.)

Definition 1 A *process object* consists of four components $(P, \hat{P}, C, M)$, where

1. P, the logic component, is an NSM.
2. $\hat{P}$, the "physical part" of the logic component, is the sub-NSM of P consisting of only "real" transitions.
3. C, the output (actuator) component, is an equivalence relation on $\overline{\Sigma}$ representing a control mask.
4. M, the input (sensor) component, is an equivalence relation on $\overline{\Sigma}$ representing an observation mask.

Given $\sigma \in \overline{\Sigma}$, we use $C(\sigma)$ and $M(\sigma)$ to denote the corresponding equivalence classes containing σ. An event is said to be completely uncontrollable (respectively, completely unobservable) if it belongs to the equivalence class of ϵ of mask C (respectively, of mask M). Two events are said to be control-equivalent (respectively, observation-equivalent or indistinguishable) if they belong to the same equivalence class of C (respectively, of M). If every equivalence class with the possible exception of the class of ϵ is a singleton, then we refer to the mask as a *natural projection.* For any $A \subseteq \Sigma$, we use π_A to denote the natural projection for which the equivalence class of ϵ is $\{\epsilon\} \cup (\Sigma - A)$. If $A = \Sigma$, we use I in place of π_Σ and refer to it as the *identity mask.*

Definition 2 Given a process P and a pair of control and observation masks (C, M), the *augmentation of P with respect to* (C, M), denoted P^{CM}, is obtained from P by adding for each state x transitions as follows:

1. If $(x, \sigma, x') \in P$ with $x \neq x'$ and $M(\sigma) = M(\epsilon)$, add transitions (x, σ, x) and (x, ϵ, x').

2. If $(x, \sigma_1, x_1), (x, \sigma_2, x_2) \in P$ with $x_1 \neq x_2$ and $M(\sigma_1) = M(\sigma_2)$, add transitions (x, σ_1, x_2) and (x, σ_2, x_1).

3. If $\sigma \notin \overline{\Sigma}_P(x)$, $C(\sigma) \cap \overline{\Sigma}_P(x) \neq \emptyset$ (i.e., σ is completely uncontrollable or control-equivalent to an existing event), and $M(\sigma) \cap \overline{\Sigma}_P(x) = \emptyset$ (i.e., σ is neither completely unobservable nor observation-equivalent to an existing event), add a self-loop transition (x, σ, x).

4. If $\sigma \notin \overline{\Sigma}_P(x)$, $C(\sigma) \cap \overline{\Sigma}_P(x) \neq \emptyset$, $M(\sigma) \neq M(\epsilon)$, and $M(\sigma) \cap \Sigma_P(x) \neq \emptyset$ (i.e., σ is not completely unobservable and is observation-equivalent to an existing event), add transitions $\{(x, \sigma, \delta_P(x, \sigma'))|\sigma' \in \Sigma_P(x) \cap M(\sigma)\}$.

5. If $\sigma \notin \overline{\Sigma}_P(x)$, $C(\sigma) \cap \overline{\Sigma}_P(x) \neq \emptyset$, $M(\sigma) = M(\epsilon)$, (i.e., σ is completely unobservable), add transitions $\{(x, \sigma, \delta_P(x, \sigma'))|\sigma' \in \Sigma_P(x) \cap M(\epsilon)\}$ as well as the self-loop transition (x, σ, x).

Example 1 Consider the deterministic state machine P shown in Figure 1 with event set $\Sigma = \{a, b, c, d\}$ and $L(P) = \overline{\{dd, ca, b\}}$. Suppose that the C-equivalence classes are $A_0 := \{\epsilon, b\}$, $A_1 := \{a, c\}$, $A_2 := \{d\}$, and the M-equivalence classes are $B_0 := \{\epsilon, c\}$, $B_1 := \{a, b, d\}$.

The augmented process P^{CM} is also shown in Figure 1. It is obtained from P by the following procedure:
1. Since P contains the transition (x_0, c, x_2) and $M(c) = M(\epsilon)$, transitions (x_0, c, x_0) and (x_0, ϵ, x_2) are added.
2. Since transitions on both b, d are defined in x_0 and $M(b) = M(d)$, the sets of successor states must be the same. Hence, we add transitions $(x_0, b, x_1), (x_0, d, x_3)$.

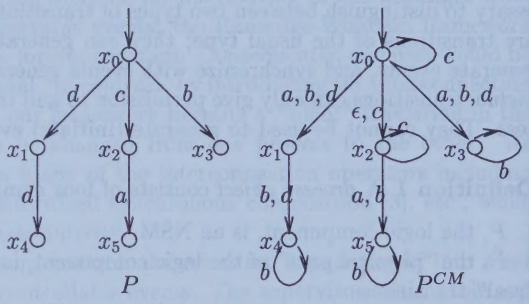


Figure 1: Diagram illustrating Example 1

3. Since c is defined in x_0 and $C(a) = C(c)$, a transition on a must be defined in x_0. Since $M(a) = M(b) = M(d)$ and there already are transitions on b, d in x_0, the new transition on a must have the same successor states x_1, x_3. Thus, we add transitions $(x_0, a, x_1), (x_0, a, x_3)$.
4. Since a is defined in x_2 and $C(a) = C(c)$, a transition on c must be defined in x_2. Since there is no event defined in x_2 that is M-equivalent to c we add the self-loop transition (x_2, c, x_2).
5. Since b is C-equivalent to ϵ, there must be a transition on b defined in every state. Thus, transitions on b must be added in states x_1, x_2, x_3, x_4, x_5. In x_3, x_4, x_5, there are no existing transitions on events that are M-equivalent to b, so we add self-loops (x_3, b, x_3), (x_4, b, x_4), (x_5, b, x_5). In state x_1, there is already a transition on d, and $M(d) = M(b)$. Thus, the new transition on b must have the same successor state as the existing transition on d. Consequently, we add the transition (x_1, b, x_4). In state x_2, there is already a transition on a, and $M(a) = M(b)$. Thus, the new transition on b must have the same successor state as the existing transition on a. Consequently, we add the transition (x_2, b, x_5).

The definition of augmentation motivates the following definition of a (C, M)-process.

Definition 3 Given a process P and pair of control and observation masks (C, M), we say that P is a (C, M)-*process* if $P^{CM} = P$.

In the remainder of this section we obtain a characterization for the set of languages which can be generated by (C, M)-processes for a given pair of control and observation masks. This leads us to introduce the notion of (C, M)-languages.

Definition 4 Given a language $K \subseteq \Sigma^*$ and a pair of control and observation masks (C, M), K is said to be a (C, M)-*language* if the following conditions are satisfied:

1. If $\sigma, \sigma' \in \overline{\Sigma}$ with $C(\sigma) = C(\sigma')$ and $s\sigma \in \overline{K}$, then $s\sigma' \in \overline{K}$.
2. If $\sigma \in \Sigma$ with $M(\sigma) = M(\epsilon)$ and $s\sigma t \in \overline{K}$, then $s\sigma^* t \subseteq \overline{K}$.
3. If $\sigma, \sigma' \in \overline{\Sigma}$ with $C(\sigma) = C(\sigma')$, $M(\sigma) = M(\sigma')$, and $s\sigma t \in \overline{K}$, then $s\sigma' t \in \overline{K}$.
4. If $\sigma, \sigma' \in \overline{\Sigma}$ with $C(\sigma) = C(\sigma')$, $M(\epsilon) = M(\sigma')$, and $s\sigma t \in \overline{K}$, then $s(\sigma')^* \sigma t \subseteq \overline{K}$.
5. If $\sigma, \sigma' \in \overline{\Sigma}$ with $C(\epsilon) = C(\sigma')$, $M(\sigma) = M(\sigma')$, and $s\sigma t \in \overline{K}$, then $s\sigma' t \in \overline{K}$.

The next lemma follows from the definitions of augmentation, (C, M)-processes, and (C, M)-language.

Lemma 1 If P is a (C, M) process, then $L(P)$ is a (C, M)-language.

Given a language K, we use $\overline{CM}(K)$ to denote the collection of all prefix closed (C, M)-superlanguages of K. Then it is clear from Definition 4 that $\overline{CM}(K)$ is nonempty and closed under arbitrary intersections and arbitrary unions. Consequently, it contains a unique infimal element, which we denote as K^{CM}.

Lemma 2 Given a nonempty language K and a pair of control and observation masks (C, M), there exists a (C, M)-process P such that $L(P) = K^{CM}$. If K is regular, then P can be chosen to be finite-state.

Proof: The construction of P starts with the deterministic Nerode construction for $\overline{K}$ and modifies it by "splitting" each state into several states and using ϵ-transitions so that events that are neither completely uncontrollable nor control-equivalent are never defined in the same state. The details are omitted.

The results of Lemmas 1 and 2 can be combined to obtain the result of the next representation theorem that describes exactly the class of languages that can be generated by (C, M)-processes. It provides the foundation for the supervisory control results in Section 4.

Theorem 1 Given a nonempty prefix closed language K and a pair of control and observation masks (C, M), there exists a (C, M)-process P such that $L(P) = K$ if and only if K is a (C, M)-language. If K is a regular (C, M)-language, then P can be chosen to be finite-state.

Remark 1 It follows from Lemma 2 that if K is regular, then so is K^{CM}. In fact if the number of states in the minimal deterministic generator for K is n, then the number of states in the generator for K^{CM} as constructed in the proof of Lemma 2 is no more than the number of control equivalence classes times n. Since the number of control equivalence classes is bounded above by the size of the event set, it follows that the NSM generator for K^{CM} has $O(n)$ states. Thus it can be verified in linear time whether a given language is a (C, M)-language. (Simply construct the NSM generator for K^{CM} and verify whether any extra transitions were added in it.)

3 Masked Composition of Process Objects

This section defines the notion of masked composition which is the interconnection mechanism of process objects we use for achieving control as well as for interaction among the components of a plant.

Definition 5 Given process objects $(P_0, \hat{P}_0, C_0, M_0)$ and $(P_1, \hat{P}_1, C_1, M_1)$, their *masked composition* (MC), denoted $(P_0, \hat{P}_0, C_0, M_0) \parallel (P_1, \hat{P}_1, C_1, M_1)$, is defined to be the process object

$$(P, \hat{P}, C, M) := (P_0^{C_0 M_0} \| P_1^{C_1 M_1},\ (\hat{P}_0 \| P_1^{C_1 M_1}) \cup (P_0^{C_0 M_0} \| \hat{P}_1),\ C_0 \| C_1,\ M_0 \| M_1),$$

where $\|$ is used both to denote the *strict synchronous composition* (SSC) of NSM's and to denote the *intersection* of equivalence relations–i.e., the equivalence relation whose equivalence classes are the intersections of the equivalence classes from the original equivalence relations. The union operation refers to sets of transitions.

A real transition on σ (i.e., a transition on σ in $\hat{P}$) can occur in the composed system provided a real transition on σ can occur in one of the component processes and σ is control-equivalent to an event that can occur in the other process. In this case, the real transition on σ in the first process is synchronized with an observation-equivalent event in the other process if one exists; otherwise the second process does not participate. For this interpretation to be accurate, by convention we regard there to be a self-loop labeled by ϵ in every state of each process. The interpretation of P (which contains the virtual as well as the real transitions) is that it describes the largest behavior possible in any composite system that has the given subsystems as components. I.e., if the 2 processes were composed with a process with $L(\hat{P}_3) = \Sigma^*$, then the real behavior of the overall system would be P.

Example 2 Consider processes $P_0 = \hat{P}_0$, $P_1 = \hat{P}_1$, $P_2 = \hat{P}_2$ where P_i can execute the event a_i and then deadlocks. Suppose that $C_1(a_0) = C_1(a_1) \neq C_1(a_2)$ and $C_2(a_0) = C_2(a_2) \neq C_2(a_1)$. The masked composition of the processes for $i = 1, 2$ has no real transitions, yet it has a virtual transition labeled by a_0. It cannot generate (initiate) this transition. However, the presence of this virtual transition allows a real transition on a_0 to occur when this process is composed with P_0. If we did not include virtual transitions in our model, then the composition of P_1 with P_2 would be the deadlock process, and associativity would not hold.

It is clear that the definitions of augmentation and masked composition extend in an obvious way to the more general situation where the control and observation masks are allowed to be state-dependent.

Example 3 Let P_0 be a deterministic process and P_1 be a nondeterministic process as shown in Figure 2.

Suppose that c is completely unobservable to P_1 but that otherwise events are completely controllable and completely observable to both processes. The NSM P resulting from the masked composition of P_0 and P_1 is shown.

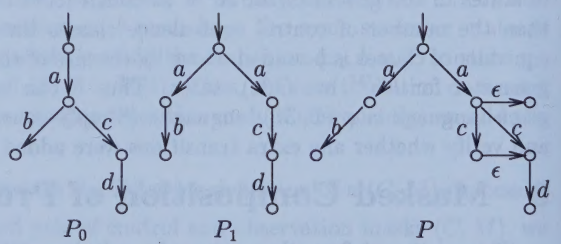


Figure 2: Diagram illustrating Example 3

Remark 2 The *prioritized synchronous composition* (PSC) [3, 11] of processes P_0 and P_1 with priority sets A and B corresponds to the special case of masked composition where $M_0 = I$, $M_1 = I$–i.e., in each process, every observation equivalence class is a

singleton (all events are completely observable to each process), and the control masks C_0, C_1 are the natural projections π_A, π_B respectively.[1]

The traditional supervisory control model [10] corresponds to the special case where $C_0 = M_0 = I$ (all events are completely controllable and completely observable to the plant), $C_1 = \pi_{\Sigma_c}$, and $M_1 = \pi_{\Sigma_o}$, where Σ_c, Σ_o denote the set of controllable events and set of observable events respectively. ∎

The next theorem proves the associativity of masked composition; it follows from associativity of SSC and intersection, and the fact that SSC distributes over "union" of conformable state machines (union refers to sets of transitions of state machines with same state space as in the definition of masked composition).

Theorem 2 Masked composition is associative:

$$\begin{aligned}[(P_0, \hat{P}_0, C_0, M_0) \parallel (P_1, \hat{P}_1, C_1, M_1)] \quad &\parallel \quad (P_2, \hat{P}_2, C_2, M_2) = \\ (P_0, \hat{P}_0, C_0, M_0) \quad &\parallel \quad [(P_1, \hat{P}_1, C_1, M_1) \parallel (P_2, \hat{P}_2, C_2, M_2)].\end{aligned}$$

The next result describes conditions under which a masked composition contains no virtual transitions.

Theorem 3 Let (P_i, P_i, C_i, M_i) $(i = 0, \ldots, n)$ be process objects with no virtual transitions, and let $(P, \hat{P}, C, M) := \|_{i=0}^{i=n}(P_i, P_i, C_i, M_i)$. Suppose $\forall \sigma \in \Sigma : \exists i$ s.t. $[C_i(\sigma) = \{\sigma\}] \wedge [M_i(\sigma) = \{\sigma\}]$. Then $\hat{P} = P = P_0^{C_0 M_0} [\!] \cdots [\!] P_n^{C_n M_n}$.

A consequence of Theorem 3 is the following corollary which states that if a system is obtained by interconnecting a finite collection of process objects, then the language of the system can be obtained by intersecting the languages of the augmented processes provided each event is completely controllable and completely observable to at least one process. This corollary can be regarded as a generalization of the language intersection result for the PSC of two processes when each event belongs to the priority set of at least one process [11, Proposition 4].

Corollary 1 Under the hypotheses of Theorem 3 the language of the composed system is given by $L(\hat{P}) = L(P) = \bigcap_{i=0}^{n} L(P_i^{C_i M_i})$.

4 Supervisory Control

This section studies the problem of control when plant as well as supervisor are represented as process objects, and are interconnected using masked composition to achieve a certain desired generated behavior.

Given a process object (P_0, P_0, C_0, M_0) representing a plant, a pair of control and observation masks (C_1, M_1) for a supervisor to be constructed, and a language K, we begin by finding necessary and sufficient conditions for there to exist a process object (P_1, P_1, C_1, M_1) (representing a supervisor) such that the generated language of the masked composition of plant and supervisor is equal to K. This is equivalent to the requirement that

$$L([P_0 [\!] P_1^{C_1 M_1}] \cup [P_0^{C_0 M_0} [\!] P_1]) = K. \tag{1}$$

[1] Actually, there is a subtle difference between PSC and this special case of MC. In PSC, it is not possible for one process to execute an event σ by itself when the other process is in a state where σ is possible after first executing an ϵ-transition. In MC, this is entirely possible.

We focus on the special case where each event is assumed to be both completely controllable and completely observable to either the plant or the supervisor so that the language intersection result of Corollary 1 holds. Thus under this assumption (1) is equivalent to $L(P_0^{C_0M_0}) \cap L(P_1^{C_1M_1}) = K$. Hence only sublanguages of $L(P_0^{C_0M_0})$ can be obtained as the closed-loop generated language. The following definition introduces the notion of realizability, which is a necessary and sufficient condition for the existence of a supervisor.

Definition 6 Given a plant (P_0, P_0, C_0, M_0), control and observation masks (C_1, M_1), and a language $K \subseteq L(P_0^{C_0M_0})$, K is said to be a *realizable* language if it satisfies the following condition: $L(P_0^{C_0M_0}) \cap K^{C_1M_1} = \overline{K}$.

It is clear from this definition that K is realizable if and only if $\overline{K}$ is realizable.

Theorem 4 Consider a plant (P_0, P_0, C_0, M_0), a pair of control and observation masks (C_1, M_1), and a language $K \subseteq L(P_0^{C_0M_0})$. Suppose $\forall \sigma \in \Sigma : [C_0(\sigma) = \{\sigma\} \wedge M_0(\sigma) = \{\sigma\}] \vee [C_1(\sigma) = \{\sigma\} \wedge M_1(\sigma) = \{\sigma\}]$. Then there exists an NSM P_1 such that the closed-loop generated language is K if and only if $K = \overline{K} \neq \emptyset$ is a realizable language. If in addition K is a regular language, then P_1 can be chosen to be finite-state.

Remark 3 It follows from Theorem 4 that the existence of a supervisor for achieving a desired generated language $K \subseteq L(P_0^{C_0M_0})$ requires that the realizability condition be verified. Since $K = \overline{K}$ is a sublanguage of $L(P_0^{C_0M_0})$, condition of realizability is equivalent to the forward containment $L(P_0^{C_0M_0}) \cap K^{C_1M_1} \subseteq K$, which is equivalent to the emptiness of the language $L(P_0^{C_0M_0}) \cap K^{C_1M_1} \cap K^c$, where $K^c := \Sigma^* - K$ is the complement of K. Let m be the number of states in the plant, and n be the number of states in the minimal deterministic generator for K. Then by definition of augmentation, the number of states in the augmented plant is still m. Also, by construction given in the proof of Lemma 2, the number of states in the nondeterministic generator for $K^{C_1M_1}$ is $O(n)$. Finally, the number of states in the generator for K^c is $n+1$. So by considering the SSC of the augmented plant NSM, the nondeterministic generator of $K^{C_1M_1}$, and the deterministic recognizer of K^c, and testing the emptiness of its recognized language, realizability can be tested in $O(mn^2)$ time. Furthermore, whenever K is realizable, the required supervisor can be chosen to be the nondeterministic generator of $K^{C_1M_1}$, which has $O(n)$ states.

Remark 4 Suppose that C_1 and M_1 are each natural projections corresponding to controllable event set Σ_c and observable event set Σ_o. Then the conditions in Definition 4 reduce to language closure with respect to the following operations:

- appending a string of uncontrollable events
- replacing an unobservable event by a nonnegative power of that event
- inserting an event that is both uncontrollable and unobservable at an arbitrary place in the string

In this case, the realizability condition reduces to standard controllability of K with respect to the language of the augmented plant together the following: Any string $t \in L(P_0^{C_0M_0})$ that can be obtained from a string $s \in \overline{K}$ by replacing each completely unobservable event σ by an arbitrary power σ^j $(j \geq 0)$ and inserting events that are both uncontrollable and unobservable must itself be in $\overline{K}$. This condition is weaker than the condition of *weak controllability and observability* used by Inan [5] in the case where $C_0 = M_0 = I$ and P_1 is not allowed to have ϵ-transitions.

Now we consider the situation where the given specification language K does not satisfy the realizability condition. Given a plant (P_0, P_0, C_0, M_0), masks (C_1, M_1), and a language K, let $\underline{P_0C_0M_0C_1M_1}(K)$ denote the collection of all realizable sublanguages of K.

Lemma 3 $\underline{P_0C_0M_0C_1M_1}(K)$ is nonempty and closed under union; consequently, it contains a unique supremal element.

When P_0, C_0, M_0, C_1, M_1 are clear from the context, we use the symbol $K^{\Uparrow}$ to denote the supremal element of $\underline{P_0C_0M_0C_1M_1}(K)$. Since K is realizable if and only if $\overline{K}$ is realizable, it follows that $K^{\Uparrow}$ is prefix closed whenever K is prefix closed. The result of the following theorem is obvious:

Theorem 5 Consider a plant (P_0, P_0, C_0, M_0), control and observation masks (C_1, M_1), and a nonempty and prefix closed language $K \subseteq L(P_0^{C_0M_0})$. Suppose $\forall \sigma \in \Sigma : [C_0(\sigma) = \{\sigma\} \wedge M_0(\sigma) = \{\sigma\}] \vee [C_1(\sigma) = \{\sigma\} \wedge M_1(\sigma) = \{\sigma\}]$. Then there exists a supervisor (P_1, P_1, C_1, M_1) such that the closed-loop generated language is contained in K if and only if $K^{\Uparrow} \neq \emptyset$.

5 Deterministic Supervisors

We briefly describe how the preceding results are modified when the supervisor process object is constrained to be deterministic. We say that the process object $(P, \hat{P}, C, M)$ is a *deterministic process object* provided P^{CM} is a deterministic state machine. A (C, M)-language K is called a *deterministic* (C, M)*-language* provided it satisfies the following condition: If $\sigma, \sigma' \in \overline{\Sigma}$ with $M(\sigma) = M(\sigma')$, and $s\sigma t$, $s\sigma \in \overline{K}$, then $s\sigma' t \in \overline{K}$. We can then show that K is a deterministic (C, M)-language if and only if it is (Σ^*, C)-controllable and (Σ^*, M)-observable, where the definition of controllability is extended to nonprojection masks in the obvious way. The class of prefix-closed deterministic (C, M)-superlanguages of K is nonempty and closed under arbitrary intersections, so an infimal element exists which we denote by K^{CMD}. We can prove that if K is nonempty, then there exists a deterministic (C, M)-process P such that $L(P) = K^{CMD}$. If K is regular, then P can be chosen to be finite-state. If K is a nonempty prefix-closed language, there exists a deterministic (C, M)-process P such that $L(P) = K$ if and only if K is a deterministic (C, M)-language. If K is a regular deterministic (C, M)-language, then P can be chosen to be finite-state.

If K is a sublanguage of $L(P_0^{C_0M_0})$, we say that K is *deterministically realizable* provided $L(P_0^{C_0M_0}) \cap K^{C_1M_1D} = \overline{K}$. We can show that K is deterministically realizable if and only if it is $(L(P_0^{C_0M_0}), C_1)$-controllable and $(L(P_0^{C_0M_0}), M_1)$-observable. Under the same assumptions as Theorem 4, we can show that there exists a deterministic supervisor such that the closed-loop generated language is K if and only if K is a nonempty prefix-closed deterministically realizable language. We emphasize that this result applies under very general conditions: (1) the plant may be nondeterministic with ϵ-transitions; (2) the plant as well as the supervisor has control and observation masks; (3) the control as well as the observation masks need not be projection-type.

In contrast to the case with realizable languages, the collection of all deterministically realizable sublanguages of K is not closed under union, so a supremal element need not exist. Consequently, there is no analogue of Theorem 5 under the added condition that the supervisor be deterministic.

6 Conclusion

We have introduced a new approach to supervisory control that views control and observation limitations of a process as the presence of a pair of masks. Such masks are

associated with supervisor as well as plant processes; thus our treatment is symmetric and suitable for modeling reactive systems. Also, controllability and observability of events are naturally treated in a unified way. The processes interact via the mechanism of masked composition in a manner that conforms to their control and observation masks. Masked composition possesses the desired property of associativity and allows the design of deterministic as well as nondeterministic processes. We have introduced the condition of realizability that is needed for the existence of supervisors, which is preserved under union. This condition is polynomially testable. Finally, the proposed framework serves as a foundation for an object oriented approach to discrete event control.

References

[1] S. Balemi, G. J. Hoffmann, P. Gyugyi, H. Wong-Toi, and G. F. Franklin. Supervisory control of a rapid thermal multiprocessor. *IEEE Transactions on Automatic Control*, 38(7):1040–1059, July 1993.

[2] M. Fabian and B. Lennartson. Petri nets and control synthesis: an object-oriented approach. In *Proceedings of the 1994 I. M. S.*, Vienna, 1994.

[3] M. Heymann. Concurrency and discrete event control. *IEEE Control Systems Magazine*, 10(4):103–112, 1990.

[4] C. A. R. Hoare. *Communicating Sequential Processes.* Prentice Hall, Inc., Englewood Cliffs, NJ, 1985.

[5] K. Inan. Nondeterministic supervision under partial observations. In Guy Cohen and Jean-Pierre Quadrat, editors, *Lecture Notes in Control and Information Sciences 199*, pages 39–48. Springer-Verlag, New York, 1994.

[6] C. P. Jobling, P. W. Grant, H. A. Baker, and P. Townsend. Object-oriented programming in control system design: a survey. *Automatica*, 30(8):1221–1261, August 1994.

[7] R. Kumar and V. K. Garg. *Modeling and Control of Logical Discrete Event Systems.* Kluwer Academic Publishers, Boston, MA, 1994. To be published.

[8] R. Kumar and M. A. Shayman. Non-blocking supervisory control of nondeterministic discrete event systems. In *Proceedings of 1994 American Control Conference*, June 1994. Submitted.

[9] R. Kumar and M. A. Shayman. Supervisory control of nondeterministic systems under partial observation. In *Proceedings of 1994 IEEE Conference on Decision and Control*, Orlando, FL, December 1994. To Appear.

[10] P. J. Ramadge and W. M. Wonham. The control of discrete event systems. *Proceedings of IEEE: Special Issue on Discrete Event Systems*, 77:81–98, 1989.

[11] M. Shayman and R. Kumar. Supervisory control of nondeterministic systems with driven events via prioritized synchronization and trajectory models. *SIAM Journal of Control and Optimization*, January 1995. To Appear.

[12] M. A. Shayman and R. Kumar. A new approach to supervisory control of discrete event systems. Technical report, University of Maryland, College Park, MD, 1994.

Why Event Observation: Observability Revisited

Changwei Cao and Feng Lin

Department of Electrical and Computer Engineering
Wayne State University, Detroit, MI 48202

Abstract

In this paper, we revisit observability defined in [11]. We show that although the original definition is intended for event observation with projection, its significance goes beyond its original intent. Specifically, we show that (1) state observation can be studied in terms of event observation; (2) event mask is mathematically equivalent to event projection; and (3) nondeterminism is caused by unobservability and can be studied in terms of observability.

1 Introduction

Since observability was introduced and studied by Lin and Wonham about ten years ago [11] [12] [13], it has played an important role in supervisory control of discrete event systems. Observability was initially defined for event observation: We distinguish, based on practical constraints, a subset of observable events from all events. Two strings look the same if they have the same sequence of observable events. Observability requires that if two strings look the same, then they must be consistent in the sense that no conflict of one event continuable after one string but not continuable after the other should occur. This definition captures precisely the existence condition for a supervisor controlling a discrete event system with partial (event) observation [11]. Recently, it was discovered that this definition of observability also describes the exact cause of nondeterminism in communicating sequential processes [14]. Because of its importance, observability has been studied extensively in the literature. For example, it was shown that observability is preserved under the union of languages but is not preserved under the intersection. Algorithms for calculating the infimal observable superlanguage of a given language and maximal observable sublanguages of a given language were also developed [1] [2] [19] [9]. These studies have lead to solutions of several important problems in discrete event systems such as supervisory control and observation problem (SCOP) [11].

As indicated, the definition of observability was initial intended for event observation only. No state observation or other observation was explicitly considered. At the first glance, this may be viewed as a drawback of the definition. However, as we will show, this is not the case. In fact, we will demonstrate that the problem with state observation can be reformulated and solved in the context of event observation.

A similar supervisory control problem was studied in [4], where the event observation is not taken as event projection as in [11], but rather as event mask. In this paper, we will formally establish the equivalence of these two approaches.

Recently, there is a renewed interest in the control of nondeterministic discrete event systems [10] [15] [20]. Although the approach sheds new light and provides new tools to the control of nondeterministic discrete event systems, we will demonstrate that the essence of the problem has not been changed and is captured by the original definition of observability.

The main contribution of this paper is summarized as follows:

- We reformulate the state observation problem and show that it can be solved in the context of event observation (Section 2).
- We reformulated the event mask problem and show that it is equivalent to event projection problem (Section 3).

- We reformulated a control problem of nondeterministic discrete event systems and show that it can be solved by using event observation on deterministic discrete event systems.

We conclude that, withstand the test of time, the original definition of observability is applicable to a large class of control problems of discrete event systems. Its importance should not be underestimated.

2 State Observation

Let us start with the supervisory control problem with state (and event) observation: Given a discrete event system modeled by an automaton

$$G = (\Sigma, Q, \delta, q_o, Q_m)$$

where, as usual, Σ is the set of events; Q the set of states; $\delta : \Sigma \times Q \to Q$ the transition function; q_o the initial state; and Q_m the set of marker states. In general, δ is only a partial function. We use $\delta(\sigma, q)!$ to denote that $\delta(\sigma, q)$ is defined. We also denote the languages generated and marked by G as $L(G)$ and $L_m(G)$ respectively. Both are subsets of Σ^*, the set of all strings of Σ, including the empty string ϵ.

Out goal is to design a supervisor so that the supervised discrete event systems achieves a control objective described by a language K.

As usual, a supervisor can control (disable) events in a controllable event set Σ_c and can observe events in an observable event set Σ_o. We denote $\Sigma_{uc} = \Sigma - \Sigma_c$ and $\Sigma_{uo} = \Sigma - \Sigma_o$. In addition to event observation, we assume that a supervisor also knows some information on the state of G: It observes $y = h(q)$, where $h : Q \to Y$ is an output mapping from the state set Q to the output set Y.

To formally define a supervisor, we extend the observation (projection) $P : \Sigma^* \to \Sigma_o^*$ to include the state observation as follows: For a given string generated by G, there is a unique sequence of states visited by the string. The supervisor observes both Σ_o and Y. Therefore, the extended projection $P_e : L(G) \to ((\Sigma_o \cup \{\epsilon\}) \times Y)^*$ is defined as

$$P_e(\epsilon) = \epsilon$$
$$P_e(s\sigma) = P_e(s)(P\sigma, h(\delta(s\sigma, q_o))).$$

Since $P_e(s)$ is the observation by a supervisor when s occurs in G, a supervisor is characterized by a mapping (the subscript s for state observation)

$$\gamma_s : P_e(L(G)) \to 2^{\Sigma}.$$

Here $\gamma_s(P_e(s))$ is interpreted as the set of events enabled by γ_s after observing $P_e(s)$. Because uncontrollable events cannot be disabled, $\Sigma_{uc} \subseteq \gamma_s(P_e(s))$.

We also use γ_s to denote the supervisor characterized by γ_s. The language generated by γ_s, $L(G, \gamma_s)$, is defined in a usual way:

(*i*) $\epsilon \in L(G, \gamma_s)$
(*ii*) $(\forall s \in L(G, \gamma_s))s\sigma \in L(G, \gamma_s) \Leftrightarrow s\sigma \in L(G) \wedge \sigma \in \gamma_s(P_e(s))$.

The language marked by γ_s is defined as

$$L_m(G, \gamma_s) = L(G, \gamma_s) \cap L_m(G).$$

γ_s is said to be nonblocking if $\overline{L_m(G, \gamma_s)} = L(G, \gamma_s)$.

Our goal is to synthesize a nonblocking supervisor γ_s such that $L_m(G, \gamma_s) = K$.

To find an existence condition for such a supervisor γ_s, we modify G by extending its states and events as follows:

$$G_e = (\Sigma \cup Y, Q \cup Q', \delta_e, q_o', Q_m),$$

where Q' is the "double" of Q, that is, for each $q \in Q$, there is a double q' corresponding to q and Q' is the set of all doubles; $\delta_e : (\Sigma \cup Y) \times (Q \cup Q') \to (Q \cup Q')$ is the extended transition function defined as follows:

$$\begin{array}{ll} \delta_e(\sigma, q) = \delta(\sigma, q)' & \text{if } \sigma \in \Sigma, q \in Q \\ \delta_e(y, q) = q & \text{if } y = h(q), q \in Q \\ \delta_e(y, q') = q & \text{if } y = h(q), q' \in Q', \end{array}$$

and everything else is undefined. This modification is illustrated in Figure 1.

To find the language generated and marked by G_e, let us define $e : \Sigma^* \to (\Sigma \cup Y)^*$ as follows:

$$\begin{array}{l} e(\epsilon) = \{\epsilon\} \\ e(s\sigma) = e(s)h(\delta(s, q_o))h(\delta(s, q_o))^*\sigma. \end{array}$$

We can now prove the following lemma.

Lemma 1

$$\begin{array}{ll} (i) & L(G_e) = \overline{e(L(G))} \\ (ii) & T_e L(G_e) = L(G) \\ (iii) & L_m(G_e) = e(L_m(G)). \end{array}$$

where $T_e : (\Sigma \cup Y)^* \to \Sigma^*$ is the projection.

The desired language K can then be extended to K_e as

$$K_e = T_e^{-1}\overline{K} \cap L_m(G_e).$$

Lemma 2 Assume that $K = \overline{K} \cap L_m(G)$. Then

$$\begin{array}{ll} (i) & \overline{K_e} = T_e^{-1}\overline{K} \cap L(G_e) \\ (ii) & K_e = \overline{K_e} \cap L_m(G_e) \\ (iii) & T_e\overline{K_e} = \overline{K}. \end{array}$$

Controllability and observability of K_e are defined in the usual way as follows:

Definition 1 K_e is controllable (with respect to $L(G_e)$ and Σ_{uc}) if

$$(\forall w \in \overline{K_e})(\forall \sigma \in \Sigma_{uc}) w\sigma \in L(G_e) \Rightarrow w\sigma \in \overline{K_e}.$$

To simplify the notation we also use P to denote the projection from $(\Sigma \cup Y)^*$ to $(\Sigma_o \cup Y)^*$.

Definition 2 K_e is observable (with respect to $L(G_e)$ and P) if

$$(\forall w, w' \in (\Sigma \cup Y)^*) Pw = Pw' \Rightarrow \mathit{consis}(w, w'),$$

where $\mathit{consis}(w, w')$ is ture if and only if

$$(\forall \sigma \in \Sigma \cup Y) w\sigma \in \overline{K_e} \wedge w'\sigma \in L(G_e) \wedge w' \in \overline{K_e} \Rightarrow w'\sigma \in \overline{K_e}.$$

Let K be a nonempty language satisfying $K = \overline{K} \cap L_m(G)$ that describes the control objective. The existence condition for γ_s can now be stated as follows.

Theorem 1 There exists a nonblocking supervisor $\gamma_s : P_e(L(G)) \to 2^{\Sigma}$ such that $L_m(G, \gamma_s) = K$ if and only if K_e is controllable and observable.

Since the proof of the above theorem is constructive, we can synthesize γ_s by first synthesizing γ_e and then letting $\gamma_s(t) = \gamma_e(Pw) - Y$.

Since state observation will provide more information, we expect that the existence condition derived in this paper be weaker than the existence condition for a supervisor with event observation only. This is proved in the following propositions.

Proposition 1 K is controllable if and only if K_e is controllable.

Proposition 2 If K is observable then K_e is observable.

Example 1 This counterexample shows that K_e is observable does not imply K is observable.

Let $\Sigma = \{a, b, c\}$ with $\Sigma_o = \{c\}$.

Let G be the generator in Figure 2 with the transition c labeled $\times$ included. The output map h is also shown in the figure. Clearly,

$$L(G) = \overline{ac + bc}.$$

Let

$$K = \overline{ac + b}.$$

That is, the transition c labeled $\times$ in the figure is excluded.

K is not observable because let $s = a, s' = b, \sigma = c$, we have

$$Ps = Ps' \wedge s\sigma \in \overline{K} \wedge s' \in \overline{K} \wedge s'\sigma \in L(G) \wedge s'\sigma \notin \overline{K}.$$

However, if we extend $L(G)$ and K to $L(G_e)$ and K_e as shown in Figure 3, then K_e is observable.

Proposition 3 If K_e is observable and $|Y| = 1$ then K is observable.

3 Event Mask

In [4], a supervisory control problem similar to that of [11] was studied using event mask instead of event projection. The event mask is defined to be

$$M : \Sigma \to \Delta \cup \{\epsilon\},$$

where Δ is the set of (event) output. That is, whenever an event σ occurs in G, a supervisor can observe $d = M(\sigma)$. If $M(\sigma_1) = M(\sigma_2)$, then σ_1 and σ_2 look the same. If $M(\sigma) = \epsilon$, then σ cannot be detected. The mapping M is extended to strings and hence $M(L(G))$ is well defined.

A (mask) supervisor is then characterized by a mapping (the subscript m for mask)

$$\gamma_m : M(L(G)) \to 2^{\Sigma},$$

with the usual requirement of $\Sigma_{uc} \subseteq \gamma_m(M(s))$. The languages generated and marked by γ_m are defined in the usual way and so is the condition of nonblocking.

The objective is again to synthesize a nonblocking supervisor γ_m such that $L_m(G, \gamma_m) = K$ for a given nonempty K satisfying $K = \overline{K} \cap L_m(G)$. To state an existence condition, we define observability with respect to M as

Definition 3 K is observable (with respect to $L(G)$ and M) if

$$(\forall s, s' \in \Sigma^*) Ms = Ms' \Rightarrow consis(s, s'),$$

where $consis(s, s')$ is ture if and only if

$$(\forall \sigma \in \Sigma) s\sigma \in \overline{K} \wedge s'\sigma \in L(G) \wedge s' \in \overline{K} \Rightarrow s'\sigma \in \overline{K}.$$

In [4], a necessary and sufficient condition is obtained for the existence of a supervisor in terms of the above definition.

Theorem 2 There exists a nonblocking supervisor $\gamma_m : M(L(G)) \rightarrow 2^{\Sigma}$ such that $L_m(G, \gamma_m) = K$ if and only if K is controllable and observable with respect to M.

We will present an alternative way to solve the same problem without using the mask. To this end, we define a substitution operator $f : \Sigma \rightarrow \Sigma\Delta$ as

$$f(\sigma) = \sigma M(\sigma).$$

f can be easily extended to strings. Therefore, $f(L(G))$ is well defined. The generator G_f for the resulting language can be easily obtained from G as illustrated in Figure 4. That is, $L(G_f) = \overline{f(L(G))}$ and $L_m(G_f) = f(L_m(G))$. We can then exteded the desired language K as

$$K_f = T_f^{-1}\overline{K} \cap L_m(G_f),$$

where $T_f : (\Sigma \cup \Delta)^* \rightarrow \Sigma^*$ is the projection.

Lemma 3 Assume that $K = \overline{K} \cap L_m(G)$. Then

(*i*) $T_f L(G_f) = L(G)$
(*ii*) $\overline{K_f} = T_f^{-1}\overline{K} \cap L(G_f)$
(*iii*) $K_f = \overline{K_f} \cap L_m(G_f)$
(*iv*) $T_f \overline{K_f} = \overline{K}$.

In the following, we will also use P to denote the projection from $(\Sigma \cup \Delta)^*$ to $(\Sigma_o \cup \Delta)^*$. An alternative way to state the existence condition is as follows.

Theorem 3 There exists a nonblocking supervisor $\gamma_m : M(L(G)) \rightarrow 2^{\Sigma}$ such that $L_m(G, \gamma_m) = K$ if and only if K_f is controllable and observable with respect to P.

Since the above proof is constructive, we can synthesize γ_m by first synthesizing γ_f and then letting $\gamma_m(t) = \gamma_f(Pw) - \Delta$.

Corollary 1 K is controllable if and only if K_f is controllable.

Corollary 2 K is observable with respect to M if and only if K_f is observable with respect to P.

4 Nondeterministic Systems

In this section, we discuss supervisory control of nondeterministic discrete event systems. There is now a new interest in nondeterministic systems [10] [15] [20], because in some applications, deterministic models are not adequate. For example, consider a discrete event system modeled by the deterministic automaton in Figure 5. If the events c and d represent the internal underlying dynamics undetectable from the outside, then from the outside, the system is modeled by the nondeterministic automaton in Figure 5. This system cannot be accurately described by the language generated by the system $L(G) = \overline{ab}$, because it does not account the fact that the system may not be able to execute any event after a. Hence we cannot simply convert the nondeterministic automaton in Figure 5 into a deterministic one and proceed as usual.

We will show that the above problem of nondeterminism is closely related to observability. To do this, we need to first answer the following question: What is nondeterminism? The answer is not as simple as one may think. For example, the nondeterministic automaton in Figure 6 does not represent a nondeterministic system because it can be easily converted to a deterministic automaton as illustrated in Figure 6. To define nondeterminism, we will use the definition in [7] [8]. To this end, we define the refusals of G_o after a string $s \in L(G_o)$, $refusals(G_o, s)$, as the set of all $X \subseteq \Sigma$ such that it is possible that G_o cannot execute any event in X after s. If G_o is represented by a nondeterministic automaton

$$G_o = (\Sigma \cup \{\epsilon\}, Q, \delta_o, Q_m)$$

with $\delta_o : (\Sigma \cup \{\epsilon\}) \times Q \to 2^Q$ a nondeterministic transition function, then

$$refusals(G_o, s) = \{X \subseteq \Sigma : (\exists q \in \delta_o(s, q_o))(\forall \sigma \in X)\delta_o(\sigma, q) \text{ is not defined}\}.$$

We now have the following definition, which is adopted from [8].

Definition 4 G_o is deterministic if

$$(\forall X \subseteq \Sigma)(\forall s \in L(G_o))(X \in refusals(G_o, s) \Leftrightarrow X \cap (L(G_o)/s) = \emptyset,$$

where $L(G_o)/s = \{t \in \Sigma^* : st \in L(G_o)\}$ and $\emptyset$ is the empty set.

It was shown in [14] that nondeterminism is caused by unobservability. To present this result, we denote the set of internal events by Σ_{in} and let $P_{in} : (\Sigma \cup \Sigma_{in})^* \to \Sigma^*$ be the projection. Consider a system whose dynamics including the internal dynamics is described by

$$G_{in} = (\Sigma \cup \Sigma_{in}, Q, \delta_{in}, q_o, Q_m),$$

where $\delta_{in} : (\Sigma \cup \Sigma_{in}) \times Q \to Q$ is the (deterministic) transition function.

We define observability of G_{in} as follows.

Definition 5 G_{in} is observable (with respect to $(\Sigma \cup \Sigma_{in})^*$, Σ and P_{in}) if

$$(\forall w, w' \in (\Sigma \cup \Sigma_{in})^*)Pw = Pw' \Rightarrow consis(w, w'),$$

where $consis(w, w')$ is ture if and only if

$$(\forall \sigma \in \Sigma)w\sigma \in L(G_{in}) \wedge w' \in L(G_{in}) \Rightarrow w'\sigma \in L(G_{in}).$$

Let G_o be the nondeterministic automaton obtained by replacing transitions in G_{in} labeled by $\sigma \in \Sigma_{in}$ with ϵ. Then the following result was proved in [14]

Theorem 4 G_o is deterministic if and only if G_{in} is observable.

Using this result, the supervisory control problem of nondeterministic discrete event systems can be studied as follows.

CASE 1: The internal dynamics of the system is known. That is, G_{in} is available in design. In this case, the design of a supervisor can be done based on G_{in} just like before. We should however include $\sigma \in \Sigma_{in}$ in the unobservable event set.

CASE 2: The internal dynamics of the system is unknown to the designer. That is, only G_o is available in design. There are four possibilities in this case.

1. Both G_o and H_o are deterministic.
2. G_o is deterministic but H_o is not.
3. H_o is deterministic but G_o is not.
4. Both G_o and H_o are nondeterministic.

Here H_o is the automaton (on Σ) describing the desired behavior of the system.

The first problem can be solved as before. The other three problems are open. We believe that the key to the solution is to find a way to "recover" the internal dynamics by inserting internal events Σ_{in} properly.

References

[1] N. Ben Hadj-Alouane, S. Lafortune and F. Lin, 1993. Control of partially observed discrete event systems with variable lookahead maximal policies. *Proceedings of the 31st Annual Allerton Conference on Communication, Control, and Computing*, pp. 898-907.

[2] H. Cho and S. I. Marcus, 1989. Supremal and maximal sublanguages arising in supervisor synthesis problems with partial observations. *Mathematical Systems Theory 22*, pp. 177-211.

[3] S. L. Chung, S. Lafortune and F. Lin, 1992. Limited lookahead policies in supervisory control of discrete event systems. *IEEE Transactions on Automatic Control, 37(12)*, pp. 1921-1935.

[4] R. Cieslak, C. Desclaux, A. Fawaz and P. Varaiya, 1988. Supervisory control of discrete-event processes with partial observations. *IEEE Transactions on Automatic Control, 33(3)*, pp. 249-260.

[5] M. Heymann, 1990. Concurrency and discrete event control. *IEEE Control Systems Magazine, 10(4)*, pp. 103-112.

[6] M. Heymann and F. Lin, 1994. On-line control of partially observed discrete event systems. *Discrete Event Dynamic Systems: Theory and Applications, 4(3)*, to appear.

[7] C. A. R. Hoare. Communicating Sequential Processes. *Comm. ACM, 21(8)*, (1978), 666-677.

[8] C. A. R. Hoare. *Communicating Sequential Processes*, Prentice-Hall, New York, (1985).

[9] R. Kumar, 1993. Formulas for observability of discrete event dynamical systems. *Proceedings of 1993 Conference on Information Sciences and Systems*, pp. 581-586.

[10] R. Kumar and M. A. Shayman, 1993. Non-blocking supervisory control of nondeterministic systems via prioritized synchronization. *Technical Research Report, T.R. 93-58*, Institute for Systems Research, University of Maryland.

[11] F. Lin and W. M. Wonham, 1988. On observability of discrete event systems. *Information Sciences, 44(3)*, pp. 173-198.

[12] F. Lin and W. M. Wonham, 1988. Decentralized supervisory control of discrete-event systems. *Information Sciences, 44(3)*, pp. 199-224.

[13] F. Lin and W. M. Wonham, 1990. Decentralized control and coordination of discrete event systems with partial observation. *IEEE Transactions on Automatic Control, 35(12)*, pp. 1330-1337.

[14] F. Lin and Z. H. Lin, 1994. Nondeterminism and Observability. *Theoretical Computer Science*, submitted.

[15] A. Overkamp, 1994. Supervisory control for nondeterministic systems. *Proccedings of 11th International Conference on Analysis and Optimization of Systems*, pp. 59-65.

[16] R. J. Ramadge and W. M. Wonham, 1987. Supervisory control of a class of discrete event processes. *SIAM J. Control and Optimization, 25(1)*, pp. 206-230.

[17] P. J. Ramadge and W. M. Wonham, 1989. The control of discrete event systems. *Proceedings of IEEE, 77(1)*, pp. 81-98.

[18] K. Rudie and W. M. Wonham, 1992. Think globally, act locally: decentralized supervisory control. *IEEE Transactions on Automatic Control, 37(11)*, pp. 1692-1708.

[19] K. Rudie and W. M. Wonham, 1990. The infimal prefix-closed and observable superlanguage of a given language. *Systems & Control Letters, 15*, pp. 361-371.

[20] M. Shayman and R. Kumar, 1994. Supervisory control of nondeterministic systems with driven events via prioritized synchronization and trajectory models. SIAM Journal of Control and Optimization, to appear.

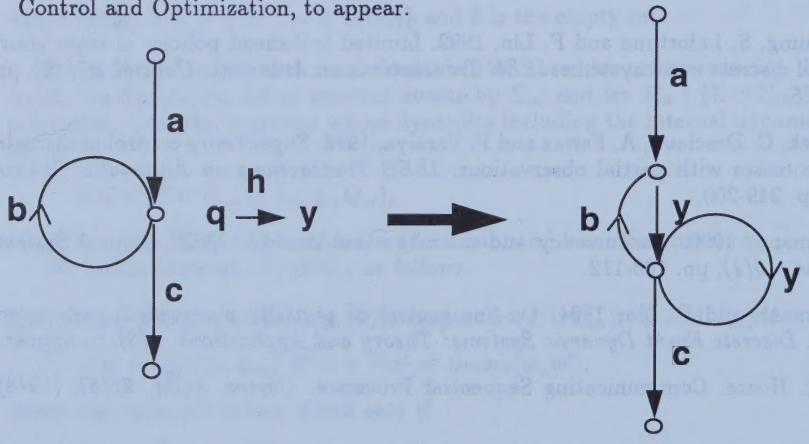


Figure 1

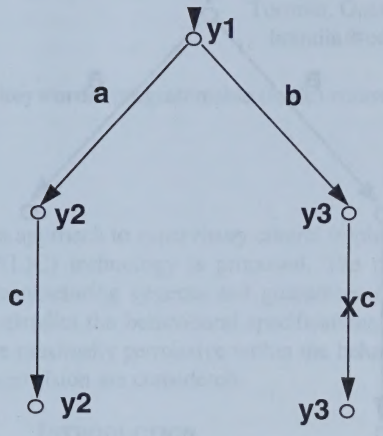


Figure 2

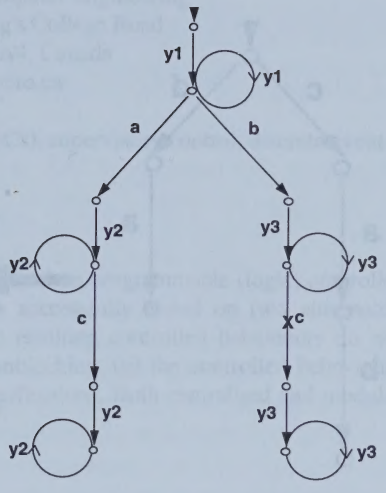


Figure 3

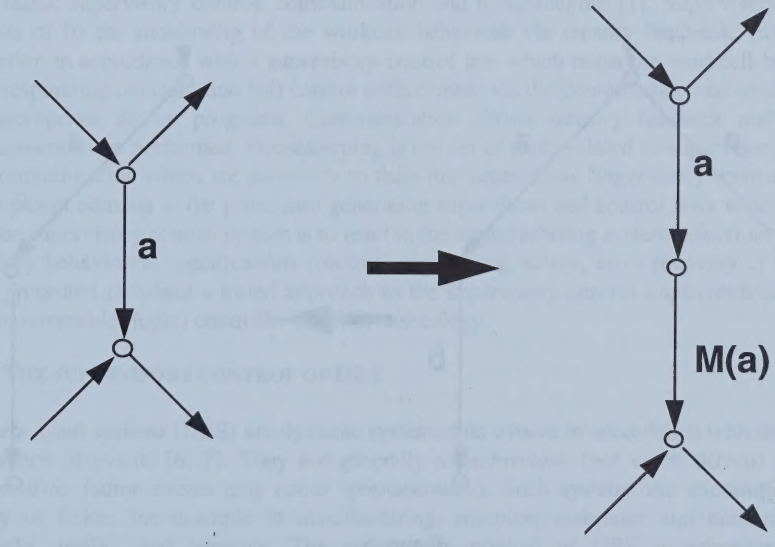


Figure 4

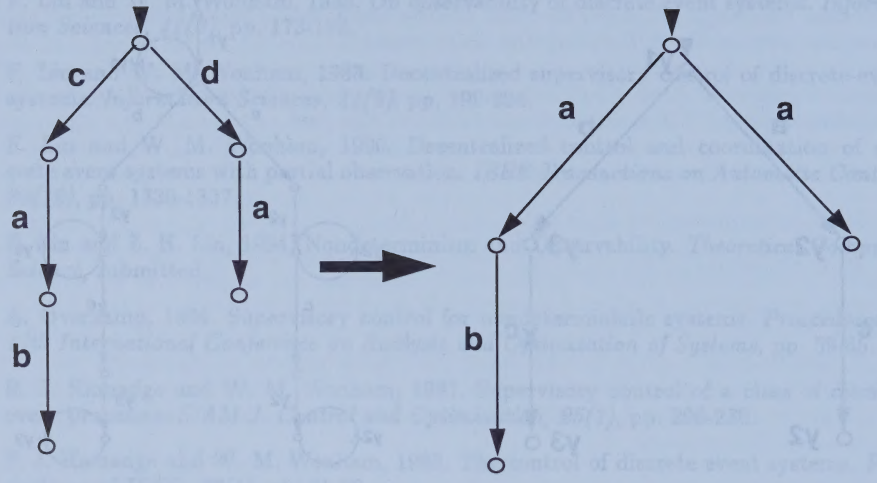


Figure 5

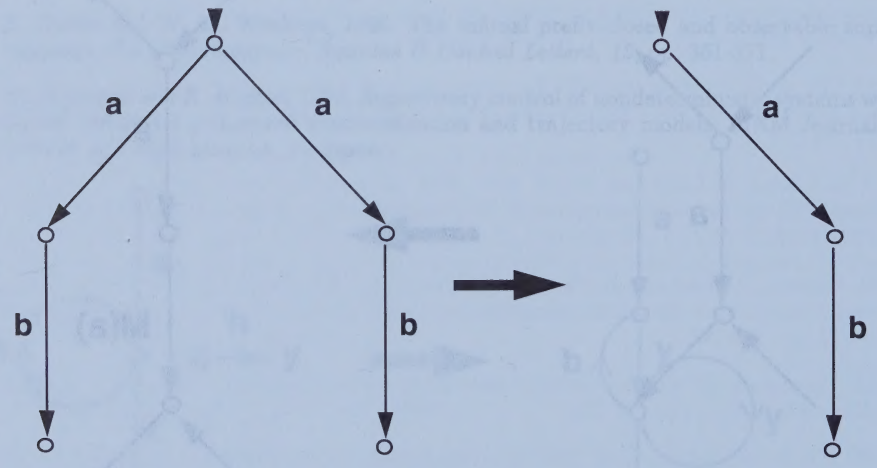


Figure 6

P(L)C-BASED SUPERVISORY CONTROL OF DISCRETE EVENT SYSTEMS

Bertil A. Brandin
Department of Electrical and Computer Engineering
University of Toronto, 10 King's College Road
Toronto, Ontario M5S 1A4, Canada
brandin@control.utoronto.ca

keywords: programmable (logic) controllers (P(L)Cs), supervisory control, discrete event systems.

ABSTRACT

An approach to supervisory control implementation based on programmable (logic) controller (P(L)C) technology is proposed. The method was successfully tested on two automated manufacturing systems and guarantees that: (i) the resulting controlled behaviours do not contradict the behavioural specifications and are nonblocking, (ii) the controlled behaviours are maximally permissive within the behavioural specifications. Both centralised and modular supervision are considered.

1 INTRODUCTION

The supervisory control system of automated manufacturing systems performs the following three tasks: supervisory control, communication and housekeeping [1]. Supervisory control consists of (i) the monitoring of the workcell behaviour via sensory feedback, (ii) control evaluation in accordance with a supervisory control law which maps the workcell behaviour to corresponding controls, and (iii) control enforcement via the downloading and execution of the appropriate device programs. Communication allows sensory feedback and control enforcement to be performed. Housekeeping is the set of tasks related to supervisory control and communication which are necessary to their implementation. Supervisory control system development consists in the procedure generating supervisors and control laws which specify how the supervisory control system is to react to the manufacturing system behaviour in order to satisfy behavioural specifications (routing, sequencing, safety, error recovery...) [1]. The work presented proposes a tested approach to the supervisory control implementation based on programmable (logic) controller (P(L)C) technology.

2 THE SUPERVISORY CONTROL OF DES

Discrete event systems (DES) are dynamic systems that evolve in accordance with the abrupt occurrence of events [6, 7]. They are generally *asynchronous* (not clock driven) and *non deterministic* (some events may occur spontaneously). Such systems are encountered in a variety of fields, for example in manufacturing, robotics, computer and communication networks, traffic, and logistics. The supervisory control of DES in accordance with behavioural specification is a new research area which is receiving increasing recognition. The approach is based on information feedback on the occurrence of events and controlled automata concepts. It offers important advantages over other approaches [1, 6, 9]:

- controlled behaviours are guaranteed not to contradict the behavioural specifications and to be nonblocking;
- controlled behaviours are optimal: maximally permissive within the behavioural specifications; and

• modular supervisors are guaranteed to be nonconflicting.

3. SUPERVISORY CONTROL SYSTEM DEVELOPMENT

The development of a supervisory control system is divided into three main steps.

3.1 THE MODELLING OF THE PLANT AND BEHAVIOURAL SPECIFICATIONS

Both the plant physical behaviour and the behavioural specifications are translated in the form of automata. The behaviour of DES, such as manufacturing systems, are modelled naturally by finite state automata. An automaton is a five-tuple $G=(\Sigma, Q, \delta, Q_m, q_o)$ where Σ represents a finite set of transitions or event labels (events are subdivided into *controllable* events, i.e. the events which may be disabled through control, and *uncontrollable* events, i.e. the events which may not be disabled through control), Q represents a finite set of states, δ is a transition function describing state transitions, Q_m represents a finite set of *marker* states (states with a particular meaning from a control perspective), and qo represents the automaton's initial state [6, 7]. Finite state automata are naturally described by directed transition graphs. In order to represent an automaton $G=(\Sigma, Q, \delta, Q_m, q_o)$, the set of states Q is identified with the nodes (represented by ∘) of a directed graph whose edges are labelled with transition labels in Σ (represented by $\circ \xrightarrow{a} \circ$). In the following, ($\longrightarrow\circ$) indicates the automaton's initial state, and (•) indicates a marker state. Modelling examples are provided in Section 4.

3.2 SUPERVISOR AND CONTROL LAW SYNTHESIS

Taking into account the supervisory control system architecture adopted (for example centralised [6] or modular [2, 3, 8]), the automata representing the physical behaviour of the system to be controlled and the automata representing the corresponding specifications are fed to a computer program which will (i) tell us whether it is possible for the system to behave within the specifications, and (ii) return the supervisor(s) embodying the maximally permissive controlled behaviour of the system within the latter specifications, and the corresponding control law(s) yielding such behaviour. The resulting supervisors which embody the controlled behaviour of the plant, are finite state automata (represented by directed transition graphs). Let S be a supervisor for a plant G. Let x be a state of S. The corresponding supervisory control law V maps every state x of S to a set of transitions $V(x)$, termed *control*. Transitions in $V(x)$ never contradict the specifications considered (and are therefore termed *legal*) and keep the system G within the specifications. Generally, these encompass a set of nominal behavioural specifications, e.g. part routing, a set of error recovery procedures for modelled errors, e.g. a robot grasping error, and a recovery procedure for generally unmodelled occurrences, e.g. the communication breakdown between the manufacturing system and the supervisory control system. The latter recovery procedure can take the form of a time-out requesting operator assistance, for example, in case no expected occurrence takes place during a specified time interval.

3.3 SUPERVISOR AND CONTROL LAW IMPLEMENTATION

Figure 1 illustrates the general organisation of a computer controlled system. Recall that a supervisory control system carries out mainly three tasks: supervisory control, communication and housekeeping. Supervisory control consists of (i) the monitoring of the plant behaviour via sensory feedback, (ii) control evaluation in accordance with a supervisory control law, and (iii) control enforcement via the downloading and execution of the appropriate device programs. Supervisory control is illustrated in Figure 2.

The monitoring of the plant behaviour via sensory feedback is essential in supervisory control. Monitoring requires that the supervisory control system possess the capability to communicate with the various devices constituting the plant, in order to gather the information provided by the devices' sensors. The sensory outputs of the various devices may not always be expressed in compatible terms. It may thus be necessary to translate these into high level language statements which can be treated uniformly by the supervisory system or easily understood by an operator.

Assume in the following, that the output of the plant output-devices, referred to as *event-data,* consists of a network address, and a bit pattern representing the occurred event.

Control evaluation determines which legal[1] events must be enabled, in accordance with the supervisory control law considered. Control evaluation requires that the supervisory control system possess both computing power and memory capacity, to carry out the required decision making operations, and the capability to interface with human operators if necessary.

Let S be a supervisor embodying the closed loop behaviour of a plant G. Let x be the current state of S. Control evaluation is carried out in the following way (cf. Figure 2). Through monitoring, event-data has been stored. Subsequently, the data is read and translated in suitable symbolic form, e.g. the *operation finish* (f1) event for the machine shown in Figure 4, indicating the end of a production cycle. The current state x of S is thus updated to say x'. Subsequently, the corresponding control $V(x')$ is evaluated and stored. If a number of modular supervisors S_i, i=1,...,n perform their task in parallel then their respective current states x_i are updated to x_i' and the controls corresponding to $\cap_{i=1,...,n} V_i(x_i')$ are energised. In a P(L)C or computer based implementation, the supervisors not only enable but also "force" the occurrence of controllable events. Therefore, the enablement and forcing of these events must occur if and only if these are physically possible. This translates into the following technical condition on the plant and supervisors behaviours: $L(G) \cap \bigcap_{i=1,...,n} L(S_i) \subseteq \bigcap_{i=1,...,n} L(S_i)$, with n possibly equal to 1.

The enforcement of controls when considered in detail, has further implications than just the ability to communicate with the workcell devices and vice versa; as is the case with monitoring, device integration becomes an important issue because of the generally heterogeneous nature of manufacturing systems (due, for example, to different machine makes). A successful and generally recognised approach to this issue consists of the development of high-level device programming languages and related device customised

[1]Recall that in the present context, *legal* has the meaning: which does not contradict any behavioural specification.

programming primitives [1]. This approach allows all the devices of a workcell to be programmed on a uniform basis. Control enforcement is carried out as follows. Control evaluation produces a list of events to be executed by corresponding devices. The latter list is read item by item, and each item is appropriately translated for each device into suitable control actions, e.g. device programs and corresponding execute statements, and thus transmitted to the corresponding devices, thus effectively enforcing the corresponding events.

Programmable (logic) controllers (P(L)C) are comparable in design to most computer-based control systems, but are specialised in logic-based control [4, 5]. They are designed to operate in industrial environments with wide ranges of operating conditions (ambient temperature, humidity, noise and vibrations). Their hardware and the corresponding software are designed for use by plant electricians and technicians. They are generally programmed in *ladder logic* or other comparable language (e.g. grafcet), and repetitively execute a single program, sequentially, from beginning to end. Programmable (logic) controllers have built-in input/output (I/O) modules linking them to the field devices constituting the plant; these are typically quite robust and easy to connect and replace.
The fast access memory of most programmable (logic) controllers is broadly divided in two: the first part containing the *user program* and the second containing the related *data table*. The user program generally accounts for most of the memory capacity, and consists of ladder logic or equivalent language statements relating inputs to corresponding *internals* (internals are user-program controlled `flags') and outputs. The data table contains the information required to carry out the user program; it includes information such as the status (energised/de-energised) of inputs, outputs and internals, timer and counter values, digital data from field devices inputs, etc.

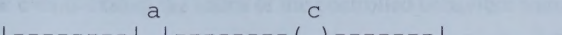
During each operating cycle, termed *scan*, the CPU sequentially reads the status of the system inputs, evaluates the control logic embodied in the user program, and consequently updates the status of the outputs by either energising or de-energising them. Data transfer is thus carried out using polling with periodic checks to determine whether the various field devices are ready for data transfer. Typically, scan times may vary from 1 ms up to 100 ms [4, 5]. Ladder logic programming is considered here. It is based on contact symbology. The five basic functions [6, 7] available on most P(L)C programming languages are described in Table 1.

function	symbol	description
energise(a)	***a*** -----( )-----	***a*** is energised if preceding logic sentence on rung is TRUE
set(a)	***a*** -----(S)-----	***a*** is energised and kept energised until reset, if preceding logic sentence on rung is TRUE once.
reset(a)	***a*** -----(R)-----	***a*** is de-energised if preceding logic sentence on rung is TRUE.
examine_on(a)	***a*** -----| |-----	returns TRUE if status of ***a*** is energised
examine_off(a)	***a*** -----| \ |-----	returns TRUE if status of ***a*** is de-energised

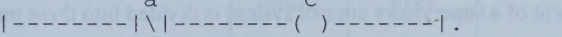
Table 1: Basic P(L)C functions

where ***a*** refers to either an input, an output or an internal for the ***examine_on*** and ***examine_off*** functions, to an output or internal for the ***energise*** function, and to an internal for the ***set*** and ***reset*** functions.

For example, for ***a***,***b*** and ***c*** referring to either an input, an output or an internal, $a \rightarrow c$ (***a*** implies ***c***) is expressed with the rung



$\neg a \rightarrow c$ ((not ***a***) implies ***c***) is expressed with the rung



We are now in a position to consider the P(L)C-based implementation of discrete-event system supervisors and the corresponding control law(s). These are coded in a Programmable (Logic) Controller (P(L)C) (or computer based control system) which will carry out feedback-based control; the supervisory control system reacts to event occurrences according to the supervisor(s) and corresponding control law(s). Recall that supervisory control consists of three tasks (i) the **monitoring** of the plant behaviour, (ii) **control evaluation**, and (iii) **control enforcement.**

Monitoring

As stated earlier, monitoring is carried out using polling: during every scan, the status of the various inputs is checked.

Control Evaluation

Let ***S*** be a supervisor for a plant ***G*** and let ***V*** be the corresponding supervisory control law. Control evaluation is carried out as follows. To every state ***x*** of ***S*** is associated an internal labelled accordingly. One such internal is energised at any time and corresponds to the supervisor's current state. Such internals are set and reset upon the occurrence of events, i.e. upon the energising of corresponding inputs. Whenever a state ***x*** of ***S*** is reached, outputs corresponding to $V(x)$ are energised. If a number of modular supervisors S_i, $i=1,...,n$ perform their task in parallel then the controls corresponding to $\cap_{i=1,...,n} V_i(x_i)$ are energised, where x_i, $i=1,...,n$ represents the state of the supervisor S_i which is currently energised.

Control Enforcement

Control enforcement is carried out by the programmable (logic) controller output modules upon the energising of given outputs.

4 Example

4.1 Problem Description

The manufacturing system considered consists of two machines and a buffer of size 1. Machine 1 and Machine 2 are connected by the buffer (cf. Figure 3). The machines are either idle, working, or down. The buffer is either empty or full. Initially the buffer is assumed to be empty and the machines are assumed to be idle. The transfer of workpieces is assumed to be part of the machines' workcycle, during which the machines pick up a workpieces upstream,

and transfer workpieces downstream.The machines operation and repair must be coordinated according to the following production and repair specifications: (i) the buffer must not overflow or underflow: Machine 1 may not start operating while a workpiece is present in the buffer, and Machine 2 may not start operating unless a workpiece is present in the buffer; and (ii) Machine 2 has repair and return to service priority over Machine 1: in case both machines are down, Machine 2 must be repaired and returned to service first.

4.2 Supervisor and Control Law Synthesis

Recall that the development of a supervisory control system is divided into three main steps:

Modelling of the plant and behavioural specifications

The manufacturing system, that is Machine 1 and Machine 2, and the behavioural specifications are modelled in the form of transition graphs as illustrated in Figures 4, 5 and 6. Consider Machine 1. Initially it is idle, its state is I1. Once it starts operating, that is the event s1 occurs, it changes state to W1. Depending on what event happens next it may either (i) change its state to idle (I1), if it successfully finishes its work cycle (f1), or (ii) change its state to down (D1), if it breaks down (b1). If the machine is down (D1), it is repaired and returned to service (r1), bringing the machine back to its idle state (I1). The same applies to Machine 2. The transfer of workpieces is assumed to be part of the machines' workcycle, so that when (s1) and (s2) have occurred, the machines are assumed to have picked up a workpiece upstream, and when (f1) or (f2) have occurred, the machines are assumed to have transferred a workpiece downstream. The events s1, s2, r1 and r2 are controllable: they can be disabled and enabled by control action. The events b1, b2, f1 and f2 are uncontrollable: they cannot be disabled and may occur spontaneously.

The buffer specification transition graph means the following: (i) initially the buffer is empty, (ii) Machine 2 may not start operating prior to the buffer being full, that is prior to the occurrence of an operation finish on Machine 1 (f1), (iii) the buffer may not be filled again (f1) before the occurrence of an operation start on Machine 2 (s2), and (iv) no other event occurrence affects the buffer specification.

The breakdown specification transition graph means the following: (i) initially Machine 2 is OK, (ii) Machine 1 may be repaired and returned to service (r1) any time as long as Machine 2 is OK, (iii) in case Machine 2 breaks down (b2), Machine 2 must be repaired and returned to service before Machine 1 may be repaired, that is the event r1 may not occur at the state KO, and (iv) no other event occurrence affects the breakdown specification.

Supervisor and control law synthesis

In order to synthesise a control law which will enforce the specifications considered, the above machine and specification transition graphs are fed to a computer program which will (i) tell us whether it is possible for the machines to behave within the specifications, and (ii) return the supervisor embodying the maximally permissive controlled behaviour of the machines within these specifications, and the corresponding control law.

In this example the machines can behave within the buffer and breakdown specifications. Centralised supervision is considered first. The corresponding supervisor embodying the maximally permissive controlled behaviour of the machines within these specifications is shown in Figure 7.

From the above transition graph the following control law is obtained by listing the controllable events exiting the states of the controlled behaviour transition graph.

superv. state x	control $V(x)$
1	s1
2	-
3	s2
4	s1
5	-
6	-
7	r2
8	s1,r2
9	r2
10	r2
11	r1
12	r1

Table 2: Control law $V(x)$

The supervisor and control law are coded in ladder logic.

Modular supervisory control is considered next. Four modules are used. Two modules, S_1 and S_2 shown in Figure 9 and Figure 6 respectively, implement the buffer and breakdown specifications respectively, the other two, S_3, and S_4, shown in Figure 4, "observe" the machines behaviours in order to guarantee that the condition $L(G) \cap \bigcap_{i=1,...,n} L(S_i) \subseteq \bigcap_{i=1,...,n} L(S_i)$, n=4, is satisfied. All modules are controllable and nonconflicting with respect to the plant. The corresponding control laws are given in Table 3. The supervisors and control laws are coded in ladder logic.

x_1	$V_1(x_1)$	x_2	$V_2(x_2)$	x_3	$V_3(x_3)$	x_4	$V_4(x_4)$
E	s1	OK	r1	I1	s1	I1	s2
F	s2	KO	r2	W1	-	W1	-
				D1	r1	D1	r2

Table 3: Control laws $V_i(x_i)$, i=1,...4

6 CONCLUSIONS

An approach to supervisory control implementation based on programmable (logic) controller (P(L)C) technology was presented. The method was successfully tested on two automated manufacturing systems and guarantees that: (i) the resulting controlled behaviours do not contradict the behavioural specifications and are nonblocking, (ii) the controlled behaviours are maximally permissive within the behavioural specifications. Both centralised and modular supervision were considered.

7 ACKNOWLEDGEMENTS

The author would like to thank Technology Ontario and Allen-Bradley (Canada) for their support of the Ontario/Rhône-Alpes Collaborative Project on Automated Manufacturing.

8 BIBLIOGRAPHY

[1] B.A. Brandin, The Supervisory Control of a Small Manufacturing Cell, SCG Report 9404, Department of Electrical and Computer Engineering, University of Toronto, April 94.

[2] B. A. Brandin, W. M. Wonham, "Modular supervisory control of timed discrete-event systems", 1993 IEEE Control and Decision Conference, San Antonio, December 1993, pp. 2230-2235.

[3] B. A. Brandin, W. M. Wonham and B. Benhabib, "Manufacturing cell supervisory control - a modular timed discrete-event system approach", 1993 IEEE International Conference on Robotics and Automation, Atlanta, May 1993, pp. 846-851.

[4] G. Michel, ``Programmable Logic Controllers - Architecture and Application", Wiley 1990.

[5] F. D. Petruzella, ``Programmable Logic Controllers", McGraw-Hill, Inc., 1989.

[6] P. J. Ramadge and W. M. Wonham, ``The control of discrete-event systems", IEEE Proceedings, Vol. 77, No. 1, January 1989, pp. 81-98.

[7] W. M. Wonham and P. J. Ramadge, ``Modular supervisory control of discrete event systems", Maths. of Control, Signals and Systems, Vol. 1, No. 1, 1988, pp. 13-30.

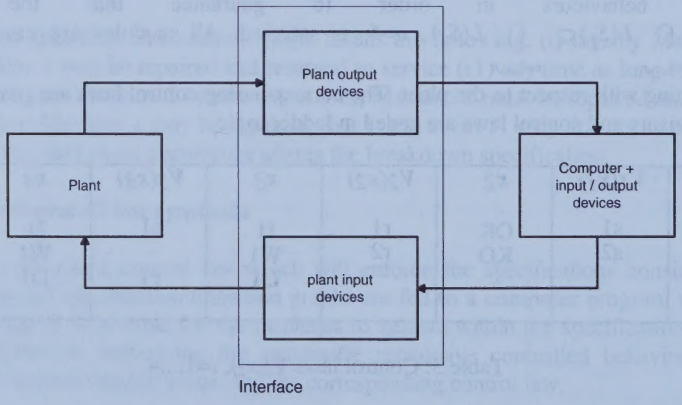


Figure 1: Computer controlled system

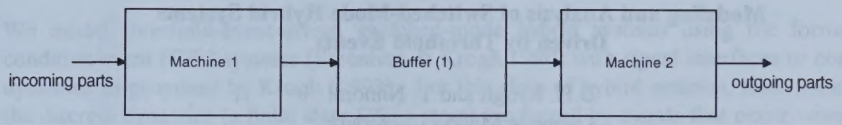


Figure 3: A small manufacturing system

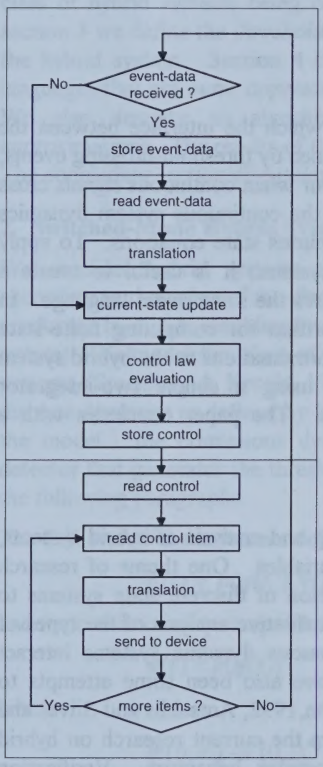


Figure 2: Supervisory control

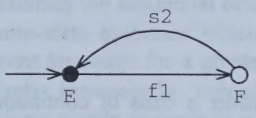


Figure 5: Buffer specification

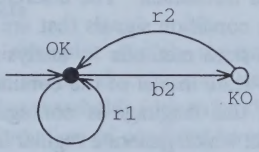


Figure 6: Breakdown specification and supervisor S_2

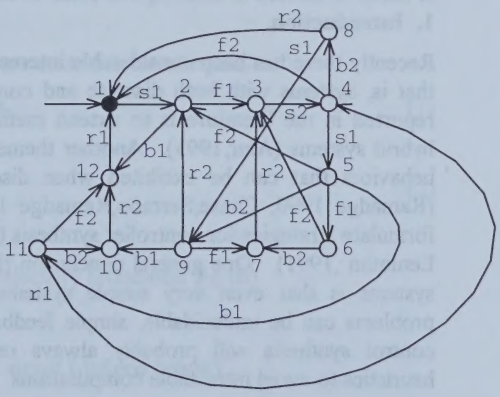


Figure 7: Supervisor S embodying the plant maximally permissive controlled behaviour

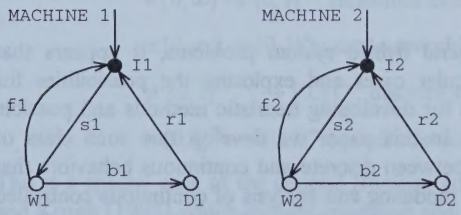


Figure 4: Machine 1 and Machine 2 (also supervisors S_3 and S_4 respectively)

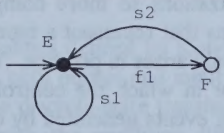


Figure 8: Buffer supervisor S_1

Modeling and Analysis of Switched-Mode Hybrid Systems Driven by Threshold Events

B.H. Krogh and T. Niinomi
Carnegie Mellon University
Dept. of Electrical and Computer Engineering
Pittsburgh, PA 15213-3890
e-mail: krogh@cmu.edu

Abstract

We consider a class of continuous-time hybrid systems in which the interface between the continuous-state dynamics and discrete-state dynamics is defined by threshold-crossing events; that is, discrete-state transitions are forced by events that occur when continuous signals cross specified thresholds. The discrete states, in turn, influence the continuous system dynamics through condition signals that are input signals to the continuous state equations. To apply discrete-state methods to analysis and synthesis of such systems, it is useful to create a discrete-state model of the continuous dynamics that generates the same event language. In general, this language is nonregular. We present an algorithm for computing finite-state automata which generate regular languages that are outer approximations to the hybrid system language. The concepts are developed and illustrated using a simple two-integrator continuous system with arbitrary rates and 0-1 thresholds. The paper concludes with a discussion of several open problems for future research.

1. Introduction

Recently there has been considerable interest in the modeling and analysis of hybrid systems, that is, systems with both discrete and continuous state variables. One theme of research reported in the literature is to extend methods for verification of discrete state systems to hybrid systems (Alur,1993). Another theme has been the qualitative analysis of the types of behaviors that can be exhibited when discrete and continuous dynamic systems interact (Ramadge 1990; Chase,Serrano,Ramadge 1993). There have also been some attempts to formulate strategies for controller synthesis (Nerode and Kohn,1992; Antsaklis and Stiver and Lemmon ,1991). One general conclusion that emerges from the current research on hybrid systems is that even very simple systems can exhibit complex behaviors. Verification problems can be undecidable, simple feedback structures can exhibit chaotic behavior, and control synthesis will probably always require the introduction of approximations and heuristics to avoid intractable computations.

Because of the difficulties that arise in general hybrid system problems, it appears that progress must be made by identifying particular cases and exploring the possibilities for analysis and synthesis as a guide to directions for developing heuristic methods and possible generalizations to more complex problems. In this paper we develop one such class of problems that represent a type of interaction between discrete and continuous behaviors that commonly arises in practice. We consider the modeling and analysis of continuous controlled behavior in which the controller switches among various ``modes" according to threshold-crossing events generated by outputs from the continuous system. Applications include gain scheduling controllers, fault-detection and recovery, responses to emergency conditions, and rule-based methods for switching controlled behaviors in response to discrete changes detected in the environment or operating constraints of the system.

We model threshold-event-driven switched-mode hybrid systems using the formalism of condition/event (C/E) systems (Sreenivas & Krogh,1990) with signal interfaces to continuous dynamics as proposed by Krogh (1993). For this class of hybrid systems, state transitions in the discrete dynamics (a finite state C/E system) are forced by events that occur when output signals from the continuous system cross specified thresholds. The output from the C/E system is a condition signal which selects the operating mode for the continuous system. The class of hybrid systems being considered is defined formally in the following section. In section 3 we define the *threshold event language* that characterizes the sequential behavior of the hybrid system. Section 4 introduces the notion of finite-state automata which accept languages that are *outer approximations* to the *threshold event language* for a given system. We also describe an algorithm for generating iteratively increasingly refined outer approximations. The results of this algorithm are illustrated for a simple example in section 5. The concluding section discussions directions for current research.

2. Switched-Mode Hybrid Systems with Threshold Events

We consider hybrid systems in which the continuous dynamics switch characteristics according to the value of the discrete state (switched mode), and the discrete state transitions are forced by events resulting from continuous signals crossing thresholds (driven by threshold events). We model this class of hybrid systems using the formalism of condition/event (C/E) systems (Sreenivas & Krogh, 1990) with the signal interfaces between the continuous and discrete dynamics as defined by Krogh (1993). Figure 1 illustrates three basic components of the model: the continuous dynamics, the C/E system dynamics, and the zero-crossing detector that generates the threshold events. Each of these components is defined in detail in the following paragraphs.

The C/E system dynamics are defined by the *discrete state transition equation*

$$\boldsymbol{q}(t) = \boldsymbol{p}(\boldsymbol{q}(t^-), \boldsymbol{v}(t)), \qquad \boldsymbol{q}(0) = \boldsymbol{q}_0 \tag{1}$$

and the *condition output equation*

$$\boldsymbol{u}(t) = \boldsymbol{r}(\boldsymbol{q}(t)) \tag{2}$$

where

$\boldsymbol{q}: [0,\infty) \rightarrow \{0,1\}^{n_q}$, discrete state trajectory (condition signal)

$\boldsymbol{q}_0 \in \{0,1\}^{n_q}$, initial discrete state

$\boldsymbol{v}: [0,\infty) \rightarrow \{0,1\}^{n_v}$, threshold event signal (defined below)

$\boldsymbol{u}: [0,\infty) \rightarrow \{0,1\}^{n_u}$, mode-switching condition signal

$\boldsymbol{p}: \{0,1\}^{n_q+n_v} \rightarrow \{0,1\}^{n_q}$, discrete state transition function

$\boldsymbol{r}: \{0,1\}^{n_q} \rightarrow \{0,1\}^{n_u}$, condition output function

This is a special case of the general Boolean condition event systems defined by Kowalewski and Krogh (1993) which include condition and event signals as both output and input signals to the C/E system. For the particular case considered here, the discrete-state transitions are deterministic, being forced by the event input signal $\boldsymbol{v}$(t), with the condition that state transitions occur only at the event instants, that is, when $\boldsymbol{v}$(t) is nonzero. This implies $\boldsymbol{p}(\boldsymbol{q},\boldsymbol{0}) = \boldsymbol{q}, \forall \boldsymbol{q} \in \{0,1\}^{n_q}$. Other properties of C/E signals and systems are discussed in (Sreenivas & Krogh, 1991).

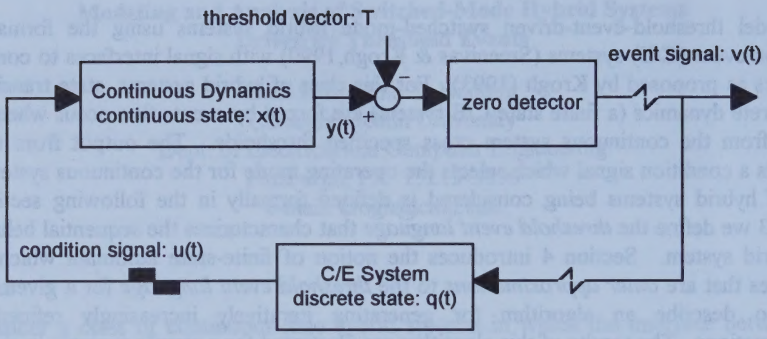


Figure 1. Block diagram of switched-mode, threshold-event-driven hybrid system

The continuous dynamics in Fig. 1 are defined by standard continuous state and output equations of the form

$$\dot{x}(t) = f_{u(t)}(x(t)), \qquad x(0) = x_0 \tag{3}$$

and

$$y(t) = g(x(t)) \tag{4}$$

where

x:$[0,\infty) \rightarrow \Re^{n_x}$, continuous state trajectory

y:$[0,\infty) \rightarrow \Re^{n_y}$, continuous output signal

f_u:$\Re^{n_x} \rightarrow \Re^{n_x}$, continuous state derivative function defined for each $u \in \{0,1\}^{n_u}$

g:$\Re^{n_x} \rightarrow \Re^{n_y}$, continuous output function

The *switched-mode behavior* of this continuous dynamics is reflected in the dependence of the continuous-state derivative on the condition signal u which selects the particular state derivative function at each point in time.

The *threshold event signal* v(t) is generated by the zero detector block in Fig. 1 which, in general, generates an event in a component of its event output signal at any instant when the corresponding component of the incoming continuous signal is zero. Since the zero detection is performed componentwise, the dimensions of the input and output signals must be the same (as is the case for the signals defined in Fig. 1). To guarantee v(t) is an event signal (any component is 1 only at isolated instants of time), we assume the incoming continuous signal never remains zero on a finite interval of time. A sufficient condition for this assumption to be true is given below. Formally, for the signals defined in Fig. 1, the input-output behavior of the zero detector is defined for each component i=1,...n_y as

$$v_i(t) = \begin{cases} 1 & \text{if } y_i(t) - T_i = 0 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where the T_i are the *thresholds* specified for each component of the output vector by the threshold vector $T \in \Re^{n_y}$. The hybrid system is said to be *threshold event driven* because it is the threshold events which cause the discrete state transitions in the C/E system (1) which in

turn cause switching in the condition signal $\boldsymbol{u}(\bullet)$ that determines the mode of behavior in the continuous dynamics (3).

From (5) it follows that each component g_i of the continuous output function $\boldsymbol{g}$, along with the threshold T_i defines a smooth manifold M_i in the continuous state space for which an event is generated every time the state trajectory "hits" the manifold. This manifold is defined by

$$M_i = \{\boldsymbol{x} \in \Re^{n_x} | g_i(\boldsymbol{x}) - T_i = 0\}.$$

When the continuous state trajectory encounters one (or more) of the manifolds, the event signal causes a deterministic transition in the state of the discrete system, which in turn switches the mode of the continuous dynamics. Since a distinct component of the event signal $\boldsymbol{v}(t)$ is associated with each manifold, the dynamic equations (3)-(4) for the C/E system identify a specific set of possible modes into which the continuous dynamics can switch on each manifold. Defining

$$U_i = \{\boldsymbol{u} \in \{0,1\}^{n_u} | \exists \boldsymbol{q}, \boldsymbol{q}' \in \{0,1\}^{n_q} \text{ and } \boldsymbol{v} \in \{0,1\}^{n_v} \text{ with } v_i = 1 \ni \boldsymbol{q} = \boldsymbol{p}(\boldsymbol{q}', \boldsymbol{v}) \text{ and } \boldsymbol{u} = \boldsymbol{r}(\boldsymbol{q})\},$$

then any event caused by the continuous state trajectory at manifold M_i will lead to a mode defined by one of the elements of U_i. This follows since U_i contains all modes generated by discrete states which are arrived at by transition triggered by an event corresponding to M_i.

To have well-defined state trajectories and condition/event signals for the class of hybrid systems defined above, we make the following assumptions:

A1. $(\boldsymbol{x}_0, \boldsymbol{q}_0) \in S_0$, where $S_0 \subseteq \Re^{n_x} \times \{0,1\}^{n_q}$ is a given *initial state relation.*

A2. There exists $\varepsilon_1 > 0$ such that $\forall i = 1, \ldots, n_v$, if $\boldsymbol{x} \in M_i$ and $\boldsymbol{u} \in U_i$, then $|\nabla_{\boldsymbol{x}} g_i(\boldsymbol{x}) \bullet \boldsymbol{f}_{\boldsymbol{u}}(\boldsymbol{x})| > \varepsilon_1$.

A3. There exists $\varepsilon_2 > 0$ such that if $\boldsymbol{x} \in M_j$ and $I(\boldsymbol{x}) = \{i | 0 < |g_i(\boldsymbol{x}) - T_i| < \varepsilon_2\}$, then $\forall \boldsymbol{u} \in U_j$ and $\forall i \in I(\boldsymbol{x})$, $(\nabla g_i(\boldsymbol{x}) \bullet \boldsymbol{f}_{\boldsymbol{u}}(\boldsymbol{x}))(g_i(\boldsymbol{x}) - T_i) > 0$.

Assumption A1 allows for a coupling between the initial state and the initial mode of the continuous dynamic system through the initial state relation, since $\boldsymbol{q}_0$ determines the initial value of the condition signal $\boldsymbol{u}$. Assumption A2 guarantees the continuous state trajectory will never remain on one of the manifolds M_i for a finite interval of time, and A3 guarantees the continuous state trajectory will never converge into an intersection of two or more manifolds. Together, assumptions A2 and A3 also guarantee there will only be a finite number of threshold events on a finite interval of time. These assertions are summarized by the following theorem.

Theorem 1. Given a hybrid dynamic system characterized by (1)-(5), if assumptions A1-A3 are true, the hybrid (continuous & discrete) state trajectory exists for a given initial condition $(\boldsymbol{x}_0, \boldsymbol{q}_0) \in S_0$, and the signal $\boldsymbol{v}(\bullet)$ is a valid event signal, that is, the components of $\boldsymbol{v}(\bullet)$ are zero except at isolated points in time, and there is only a finite number of nonzero points on a finite interval of time.

3. Threshold Event Languages

In applications, specifications and performance requirements for hybrid systems are often discrete in nature; that is, the primary properties of interest are qualitative and can be expressed in terms of discrete, rather than continuous, states. Consequently, purely discrete models are most useful for evaluating system properties or synthesizing controllers,

particularly when the controllers are discrete-state by nature, as is the case for programmable logic controllers (PLCs). In these cases, one is typically interested in the discrete behavior of the hybrid system for a whole class of continuous state trajectories, rather than the system behavior for a single trajectory.

In this paper we are interested in the discrete behavior of hybrid systems as characterized by the sequences of threshold events that are generated for given sets of initial states. To study these sequences, we map the timed sequences of threshold events characterized by the threshold event signal $v(\bullet)$ into an untimed sequence of event signals using the method introduced by Sreenivas and Krogh (1990) to generate C/E languages. Toward this end we introduce the threshold event set Θ and a mapping $V:\{0,1\}^{n_v} \rightarrow \Theta$ from the possible nonzero values of the threshold event signal into the event set. The sequence of threshold events in Θ corresponding to a particular threshold event signal $v(\bullet)$ is defined by the *event projection* P_V function which is defined as follows. Given the event signal $v(\bullet)$, let $0 \le t_1 < t_2 < \ldots$ denote the monotone increasing sequence of instants when $v(t) \neq 0$. (Note that ins general this sequence may be finite or infinite length.) The projection of the threshold event signal is then given by

$$P_V(v(\bullet)) = V(v(t_1))V(v(t_2))\cdots. \tag{6}$$

To avoid having to deal with a mixture of finite-length and infinite-length sequences of threshold events being generated by a given hybrid system, we make the following two-part assumption:

A4. Given a hybrid system (1)-(5) satisfying A1-A3: (a) from any initial state in S_O, the continuous state trajectory reaches some manifold M_i in finite time; and (b) for any $i=1,\ldots,n_v$, if $x(t) \in M_i$ for some $t>0$, and $u \in U_i$, then there exists some $i'=1,\ldots,n_v$ and $t'>t$ such that $x(t') \in M_i$.

Assumption A4 indicates that threshold events will occur indefinitely, which means that the event sequence associated with the threshold event signal will be infinite in length. We assume A4 throughout the remainder of this paper. Consequently, the projection mapping P_V maps threshold event signals $v(\bullet)$ into elements of Θ^ω, the set of infinite-length sequences of threshold events. Methods for testing assumption A4 for a given hybrid system are discussed briefly in the concluding section.

We can now identify a formal language with a given hybrid system by considering all possible threshold event strings that can result from a given set of initial states. In particular, suppose we are interested in characterizing all possible behaviors of the hybrid system starting with any initial state in a given set $S \subseteq S_O$. We let $L_S \subseteq \Theta^\omega$ denote the *threshold event language* for the system defined as the set of all threshold event sequences corresponding to the possible threshold event signals generated by the hybrid system starting from some initial state in S. The sequential behaviors of the hybrid system can then be specified and verified in terms of the properties language L_S.

Characterizing the language L_S is extremely difficult, if not impossible, for all but the most trivial examples. This is because the discrete behavior depends, in general, on the evolution of the continuous state trajectories, and computations of sets of these trajectories are wrought with numerical difficulties, in addition to being numerically intensive. Even when the continuous dynamics are trivial, requiring no numerical integration *per se*, as is the case with pure integrator dynamics, it becomes extremely difficult to represent accurately the evolution of the sets of state trajectories. But even if the numerical issues could be solved, perhaps by using symbolic computations or other sophisticated methods, the fact is that the language L_S

can be nonregular, making it impossible to characterize it with a finite state generator. (An example of such a case is given in section 5.)

Given the above difficulties, one should evaluate whether a complete characterization of L_S is, in fact, required. In many situations in discrete and hybrid systems, it is often the case that to verify the properties of interest or synthesize a desired system, one does not require a complete, detailed representation of all possible behaviors of the system. One can often settle for a very imprecise representation of the system dynamics provided it includes all critical behaviors. If a property holds for these critical behaviors, plus perhaps some other behaviors that are modeled but may never occur, that is sufficient to verify the property for the actual system. These practical considerations, combined with the extreme complexity of the behaviors of hybrid systems, lead to consideration of the concept of approximating behaviors in the following section.

4. Approximating Threshold Event Languages with Finite Automata

Given a threshold event language L_S and a language $L \subseteq \Theta^\omega$, we say L is an *outer approximation of* L_S if $L \supset L_S$. In this section we present an algorithm for obtaining a sequence of finite-state automata that accept languages (in the Buchi automata sense) that are increasingly refined outer approximations to a given threshold event language. Formally, a (Buchi) automaton B is a 5-tuple, $B = (S, \Theta, \delta, s_o, F)$, where S is a finite set of states, Θ is the finite set of (threshold) event symbols, $\delta: S \times \Theta \to 2^S$ is the partial state transition function (defined only for event symbols that are accepted at a given state), $s_0 \in S$ is the initial state, and $F \subseteq S$ is the set of *marker states*. An infinite length sequence of symbols $\sigma \in \Theta^\omega$ is accepted by the automaton B if:

(i) for $\sigma = \theta_1\theta_2\cdots, \exists s_1, s_2, \ldots \in S$ such that $\forall\, k = 1,2,\ldots,\ s_k \in \delta(s_{k-1}, \theta_k)$; and

(ii) $\exists k_1 < k_2 < \cdots$ such that $\forall\, j = 1,2,\ldots,\ s_{k_j} \in F$.

In this paper we assume $F=S$. Given an automaton B, the language of B is the set of infinite-length sequences accepted by B. We denote this language by $L(B)$. Our objective, then, is to create a sequence of automata, $B_1, B_2, \ldots$ such that for the given threshold event language L_S we have $L(B_1) \supset L(B_2) \supset \cdots \supset L_S$. In the case when there exists a finite-state automata that accepts, we would like to obtain an automaton B_n after n iterations such that $L(B_n) = L_S$.

We introduce the following definitions and notation to describe the algorithm for computing the sequences of automata described above given a hybrid system as defined in section 2.. For $i=1,\ldots,n_v$, define Q_i as the set of all discrete states reached when manifold M_i is reached, that is, $Q_i = \{\boldsymbol{q} \in \{0,1\}^{n_q} \mid \exists \boldsymbol{u} \in U_i \ni \boldsymbol{u} = \boldsymbol{r}(\boldsymbol{q})\}$. A set of (hybrid) states $R \subseteq \{\bigcup_{i=1}^{n_v} (M_i \times Q_i)\}$ is said to be an *infinitely recurring set* if for any state trajectory starting at an initial condition $(\boldsymbol{x}_0, \boldsymbol{q}_0) \in R$, there exists an infinite sequence of time instants, $0 < t_1 < \cdots$, such that $(\boldsymbol{x}(t_k), \boldsymbol{q}(t_k)) \in R$ for all $k=1,2,\ldots$ .In other words, any state trajectory starting in R passes through R infinitely often. Since we assume there are an infinite number of threshold events (A4), if follows that there will always be some infinitely recurring set as defined above. In particular, $R = \{\bigcup_{i=1}^{n_v} (M_i \times Q_i)\}$ is always an infinitely recurring set. In the following we use the partition Π of $\bigcup_{i=1}^{n_v} (M_i \times Q_i)$ defined by the sets $\pi_{\nu,\boldsymbol{q}}$, for $\boldsymbol{\nu} \in \{0,1\}^{n_v}, \boldsymbol{q} \in \{0,1\}^{n_q}$, where

$$\pi_{v,q} = \{(x,q') \in \Re^{n_x} \times \{0,1\}^{n_q} | x \in M_i \Leftrightarrow v_i = 1, q' = q\}.$$

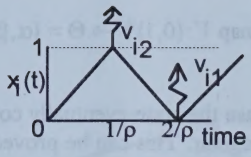
Each $\pi_{v,q}$ is a distinct subset of a manifold, or the intersections of manifolds, and an associated discrete state, where v is the threshold event signal value generated when a state trajectory hits $\pi_{v,q}$ and q is the discrete state entered when v occurs.

For a given hybrid system with threshold event language L_S, we assume the initial state set $S_0=R_0$, an infinitely recurring set. To obtain the finite state generators for outer approximations of L_S, we construct iteratively initial subtrees of an infinite tree in which each infinite-length path corresponds to a string in L_S. Each node is labeled by a triple (l,m,n), where l is the number of events generated so far from the root, m is an index on the nodes in level l of the tree, and n is the number of times an event along the current path from the root node was generated by passing through R_0. A node (l,m,n) in the tree corresponds to a threshold event generated when the continuous state trajectory encounters a particular manifold or multiple manifolds simultaneously, and the resulting nonzero value $v_{l,m,n}$ of the threshold event signal $v(\bullet)$ causes a transition into a particular discrete state $q_{l,m,n}$. Thus, each node of the tree can be associated with a particular partition subset of the manifold-discrete state set, namely $\pi_{v_{l,m,n},q_{l,m,n}}$.

The tree is generated by computing subset of the partition sets for each node that uniquely identify the subsets of continuous and discrete states that generate the particular path from the root node. These subsets, denote by $R_n^{l,m} \subseteq \pi_{v_{l,m,n},q_{l,m,n}}$, are computed as follows. The root node is associated with R_0 and no triple is assigned. The nodes for the initial level of the tree (l=0) is created for each nonempty set of the form $R_0^{0,m} = R_0 \cap \pi_{v,q}$, that is, the intersection of the initial state set R_0 with each of the partition sets is computed and a node is created (and an index m assigned) for each nonempty intersection. The corresponding threshold event vectors and discrete state values for the partition sets are associated with each node. These initial sets are the starting point for propagating forward in time the evolution of the hybrid state trajectory.

To represent the forward propagation of a set of hybrid states, we define the mapping $\Im: 2^{\Re^{n_x} \times \{0,1\}^{n_q}} \to \bigcup_{i=0}^{n_v} (M_i \times Q_i)$, where, in words, $\Im$ maps a subset of hybrid states into the subset of the manifold-discrete state set corresponding to the first encounters with this set starting with initial states in the given subset. Applying this operator to each $R_0^{0,m}$, the next level of nodes are defined by the intersections $\Im(R_0^{0,m}) \cap \pi_{v,q}$. These new nodes are labeled $R_0^{1,m}$ if they are not subsets of R_0, and they are labeled $R_1^{1,m}$ if they are subsets of R_0, with m assigned as a distinct index for each new node.

The construction of the tree continues as described above. To obtain the desired finite state generators, subtrees are computed until all terminal nodes have the same value of n. Since R_0 is an infinitely recurring set, it is guaranteed that these computations will terminate for any finite n. The n^{th} Buchi automaton B_n is defined by the state transition graph obtained from the subtree with all nodes labeled (l,m,n') where n'=0,...,n-1. The terminal nodes of this subtree are replaced by the root node, creating the state-transition graph, and each branch is labeled by a threshold event $\theta=V(v_{l,m,n})$ where (l,m,n) is the label of the terminal node for the branch. The initial branches from the root node (going to nodes $(0,m,0)$) are labeled with null events.



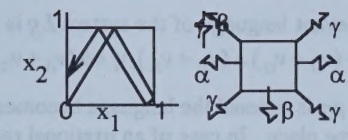


Figure 2. Integrator state trajectory. **Figure 3.** State trajectories and associated events.

Given the above construction (which is described only in words for lack of space), we have the following theorem.

Theorem 2. The languages for the sequence of Buchi automata $B_1, B_2, \ldots$ defined by the construction above are outer approximations to the threshold event language L_S. Moreover,

$$L(B_1) \supseteq L(B_2) \supseteq \cdots \supseteq L_S.$$

The construction algorithm as described above will not generate efficient representations of the generators for the outer approximating languages. Indeed, the construction above will create an unbounded sequence of Buchi automata with an unbounded number of states even if there exists a finite state generator for L_S. To compute more efficient representations of the outer approximating languages, and to define a stopping criterion for the computation that gives a finite-state representation for L_S when one exists, a backward mapping of the manifold-discrete state subsets must be computed to refine the granularity of the subset representations. The full algorithm, which has been implemented for the example in the following section, will be reported elsewhere.

5. Two-Integrator Example

We illustrate the calculation of approximating finite state machines using a simple example of hybrid system. We consider two uncoupled integrators which each integrate at a constant rate until the integrator state hits a threshold. The threshold events switch the sign of the integration rate, creating infinite cyclic behavior for each integrator. Let us distinguish the two integrators by subscript $i=1,2$. The differential equation for each integrator is $\dot{x}_i = f_{u_i}(x) = (-1)^{u_i}\rho_i$ where u_i is the condition signal supplied to the integrator i, and $\rho_i \in \Re^+$ is the rate of integration. Thus, when u_i is 0, x_i increases, and when u_i is 1, x_j decreases. Without loss of generality, we assume $\rho_1 = 1$. The output functions $g_{ij}(x)$ are defined as $g_{i1}(x) = -x_i, g_{i2}(x) = x_i - 1$. The event vector $[v_{11} \quad v_{12} \quad v_{21} \quad v_{22}]^t$ will be generated as $v_{ij} = 1 \leftrightarrow g_{ij}(x) = 0$.

The C/E system is defined by discrete state vector $q(t) = [q_1(t) \quad q_2(t)]^t$ and the state transition is defined as

$$q_i(t) = p_i(q_i(t^-), v_i(t)) = \begin{cases} 1 & if \quad (q_i(t^-) = 0) \wedge (v_{i2} = 1) \\ 0 & if \quad (q_i(t^-) = 1) \wedge (v_{i1} = 1) \\ q_i(t^-) & \text{otherwise} \end{cases}$$

The condition output of the C/E system which determines the modes of the integrators is defined as $u_i = q_i$. Figure 2 shows the state trajectory for a single integrator starting with zero initial state. When we consider two integrators, the continuous state moves in the unit square on $x_1 - x_2$ plane shown as Figure 3.

The threshold event language of the system L_S is defined by a map $V:\{0,1\}^{n_v} \to \Theta = \{\alpha,\beta,\gamma\}$ such that $\alpha \leftrightarrow (v_{11}+v_{12}) \wedge (v_{21}+v_{22}), v_i \leftrightarrow (v_{i1}+v_{i2}) \wedge \overline{(v_{j1}+v_{j2})}$.

When the rate ρ_2 is rational the language becomes regular because the state eventually comes back to the same place. In case of an irrational rate, L_S is non-regular. This can be proved by showing any different prefix will have a different postfix by considering trajectory in Figure 3 for irrational case, and the fact that any regular language must have right equivalence classes of finite index.(Nerode 1958). Thus when we choose irrational rate, we cannot obtain exact finite state model for L_S.

We have implemented a computer program which generates finite state approximation of two integrator problem. Choosing ρ_2=0.5, the program identified finite state representation of the language generator as well as set of partitions of manifolds as Figure 4. When we choose ρ_2=1.41421356, the program obtained finite state approximation of generator as Figure 5.

6. Discussion

This paper presents an algorithm for computing finite-state approximations to languages generated by threshold events in hybrid systems. Numerical implementations of this algorithm will always give approximate results. Thus, the concept of outer approximations needs to be incorporated into the computational implementation so that numerical truncation does not create incorrect representations of the reachable states. Currently we are considering the application of this method to the problem of verifying properties of threshold-driven hybrid systems.

References

[Alur,1993] Alur, R., Henzinger, T.A., Ho, P.H., Hybrid Automata: an algorithmic approach to the specification, Hybrid Systems, pp.209-229, Springer Verlag, 1993.

[Antsaklis,Stiver,Lemmon,1993] Antsaklis, P.J., Stiver, J.A., Lemmon, M., Hybrid system modeling and autonomous control systems, Hybrid Systems,Springer-Verlag; pp. 366-92,1993.

[Chase,Serrano,Ramadge,1993] Chase, C. Serrano,J. Ramadge,P.J., Periodicity and Chaos from Switched Flow Systems: Contrasting Examples of Discretely Controlled Continuous Systems, IEEE Trans. on Automatic Control, Vol. 38, No.3, p70-83,January 1993.

[Krogh, 1993] Krogh, B.H., Condition/Event Signal Interfaces for Block Diagram Modeling and Analysis of Hybrid Systems, Proceedings 1993 Symposium on Intelligent Control, pp. 180-185, Aug 1993.

[Nerode,Kohn,1992]Nerode, A. Kohn, W., Models for hybrid systems: Automata, topologies, stability. Technical Report 93-11, Mathematical Science Institute, Cornell University, March 1993.

[Nerode 1958], Linear automaton transfromations,Proc. AMS, vol 9,pp541-544,1958

[Ramadge,1990]Ramadge,P., On the Periodicity of Symbolic Observations of Piecewise Smooth Discrete-Time Systems, IEEE Trans. on Automatic Control, Vol. 35, No.7, p807-713,July 1990.

[Sreenivas,Krogh,1991] Sreenivas, R.S., Krogh, B.H., On discrete state realizations for condition/event systems, Proceedings Twenty-Eighth Annual Allerton Conference on Communication, Control and Computing Oct. 1990, pp. 465-74,1991.

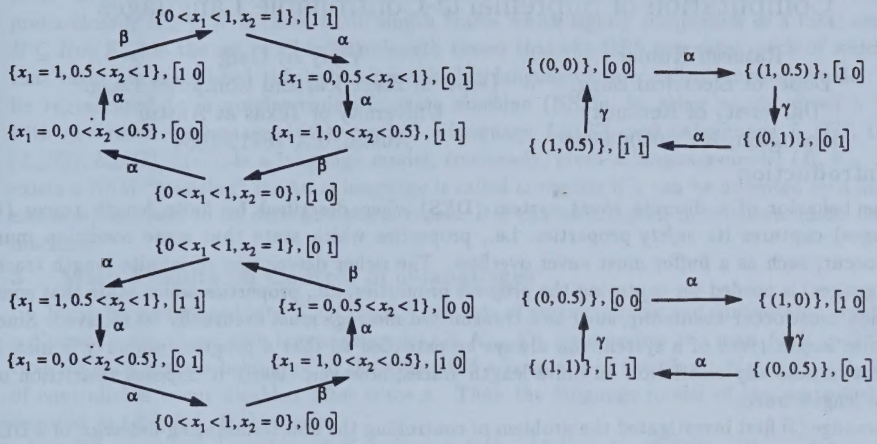


Figure 4. Exact finite state generator for two integrator example with ρ_2=0.5.

Hybrid State	Event	Discrete State	Intervals
1	γ	[0 0]	[0 0]
	γ	[0 1]	[0 1]
	γ	[1 0]	[1 0]
	γ	[1 1]	[1 1]
2	α	[0 0]	[0 1]-[0 0]
	α	[0 1]	[0 1]-[0 0]
	α	[1 0]	[1 1]-[1 0]
	α	[1 1]	[1 1]-[1 0]
3	β	[1 0]	[0.292893 1]
	β	[1 1]	[0.707107 1]
	β	[0 0]	[0.292893 0]
	β	[1 0]	[0.707107 0]
4	β	[1 1]	[0.292893 1]-[0 1]
	β	[1 1]	[1 1]-[0.707107 1]
	β	[0 0]	[0.292893 0]-[0 0]
	β	[1 0]	[1 0]-[0.707107 0]
5	β	[1 1]	[1 1]-[0.292893 1]
	β	[1 1]	[0.707107 1]-[0 1]
	β	[0 0]	[1 0]-[0.292893 0]
	β	[1 0]	[0.707107 0]-[0 0]

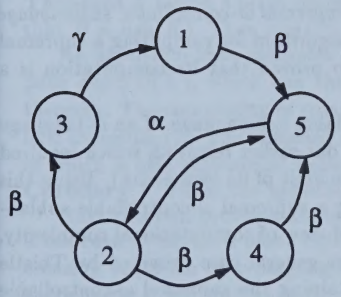


Figure 5. Finite-state generator for outer approximation language (n=1) for two-integrator example when ρ_2=1.41421356.

Computation of Supremal ω-Controllable Languages *

Ratnesh Kumar
Dept. of Electrical Engg.
University of Kentucky
Lexington, KY 40506-0046

Vijay K. Garg
Dept. of Electrical and Computer Engg.
University of Texas at Austin
Austin, TX 78712-1084

1 Introduction

The behavior of a discrete event system (DES) when described by finite length traces ($*$-languages) captures its *safety* properties, i.e., properties which state that some condition must *never* occur, such as a buffer must never overflow. The richer description of infinite length traces (ω-languages) is needed for capturing the *progress* properties, i.e., properties which state that some condition must occur *eventually*, such as a transmitted message must eventually be received. Since any finite length trace of a system can always be extended so that a progress property is met, it does not impose any restriction on finite length traces; however, clearly it imposes restriction on infinite length traces.

Ramadge [3] first investigated the problem of controlling the non-terminating behavior of a DES and showed that a supervisor for achieving a desired non-terminating behavior exists if and only if the desired behavior is *relative ω-closed* and has a controllable prefix. This property is not preserved under union in general; hence a *minimally restrictive supervisor* need not exist. This lead Thistle and Wonham [4] to introduce the notion of *ω-controllability* under which it is possible to control the non-terminating behavior so that the controlled non-terminating behavior approximates the desired one to a given accuracy. An equivalent result, in a less general setting, was independently obtained by Young, Spanjol and Garg [5], where the notion of *finite-stabilizability* was introduced as a condition for achieving the desired behavior to a given accuracy. The attractive feature of ω-controllability is that it is preserved under union so that a *supremal ω-controllable sublanguage* of a given ω-language exists. Thistle and Wonham [4] gave an algorithm for computing a supremal ω-controllable sublanguage of an ω-regular language, and also proved that its computation is a NP-complete problem.

In this note we present a formula for a supremal ω-controllable sublanguage of an ω-language that can be represented as the limit of a $*$-language, extending our earlier result [2] which required that the ω-language be ω-closed (so that it can be written as the limit of its own prefix). Using this formula we obtain a polynomial time algorithm for computing a supremal ω-controllable sublanguage when it is deterministic ω-regular. Thus from the point of view of computational complexity, this is an important special case of the algorithm for the more general case provided by Thistle and Wonham. We also provide an iterative technique for obtaining the supremal ω-controllable sublanguage without assuming ω-regularity.

2 Notation and Preliminaries

Letting Σ denote universe of events, Σ^* denotes the set of all finite length traces, and Σ^ω denotes the set of all infinite length traces. Symbol σ is used for denoting an event; s for denoting a finite length trace; K for denoting a subset of Σ^*, known as a $*$-language; B for denoting a subset of Σ^ω, known as an ω-language. The *prefix* operation on $\Sigma^* \cup \Sigma^\omega$ is denoted by $pr(\cdot)$, and the *limit* operation on Σ^* is denoted by $lim(\cdot)$. It is readily verified that

$$K \subseteq pr(K); \quad B \subseteq lim(pr(B)); \quad pr(lim(K)) \subseteq pr(K);$$

the corresponding reverse containments do not hold in general. K is said to be prefix-closed if the first reverse containment holds; B is said to be *ω-closed* if the second reverse containment holds,

*This research was supported in part by the Center for Robotics and Manufacturing, University of Kentucky, in part by the National Science Foundation under Grants NSF-CCR-9110605 and ECS-9409712, in part by a grant from IBM, and in part by a TRW faculty assistantship award.

and $lim(pr(\cdot))$ is concisely written as $clo(\cdot)$; finally, the third reverse containment holds if and only if K is *deadlock-free*, i.e., any trace of K can be extended to another trace in K [1].

The qualitative behavior of a DES can be described using a language triple (B, K_m, K), where K is the set of all finite length traces the DES can generate, consequently, it is nonempty and prefix-closed; $K_m \subseteq K$ is those finite length traces which signify completion of a task; and finally, $B \subseteq lim(K_m)$ is the set of all infinite length traces that the DES generates, each of which signify that tasks are completed infinitely often. Such a language triple, called language-model, can also be represented by a nondeterministic state machine (NSM). In other words, given a NSM G, with generated $*$-language $L(G)$, marked $*$-language $L_m(G)$, and ω-language $L_\omega(G)$, the triple $(L_\omega(G), L_m(G), L(G))$ is a language model; conversely, given a language-model (B, K_m, K) there exists a NSM "realizing" it. An ω-language is called ω-regular if it can be accepted by a finite state machine; it is said to be *deterministic ω-regular* if it can be accepted by a deterministic finite state machine.

3 Main Results on Control of ω-languages

Σ can be partitioned into $\Sigma_u \cup (\Sigma - \Sigma_u)$, the sets of uncontrollable and controllable events. Given a discrete event plant with language model (B, K_m, K), a supervisor is a map $f : K \to 2^{\Sigma - \Sigma_u}$ such that for each trace s belonging to the generated $*$-language of the plant, $f(s) \subseteq \Sigma - \Sigma_u$ is the set of controllable event disabled after trace s. Then the language model of the controlled plant is denoted as (B^f, K_m^f, K^f), where

$$[\epsilon \in K^f; \quad s \in K^f, s\sigma \in K, \sigma \notin f(s) \Rightarrow s\sigma \in K^f]; \quad [K_m^f := K_m \cap K^f]; \quad [B^f := B \cap lim(K^f)].$$

The triple (B^f, K_m^f, K) is indeed a language-model [1]. Since $B^f \subseteq lim(K_m^f)$, it follows that $pr(B^f) \subseteq pr(lim(K_m^f)) \subseteq pr(K_m^f) \subseteq K^f$. The supervisor is said to be *ω-non-blocking* if $pr(B^f) = K^f$. A necessary and sufficient condition for the existence of a ω-non-blocking supervisor that achieves a desired ω-language B with any given accuracy is ω-controllability [4]. In the following theorem we present a formula for $supC^\omega(B)$, the supremal ω-controllable sublanguage of B, which exists as ω-controllability is preserved under union. The notation $supCD(K)$ denotes the supremal controllable and deadlock-free sublanguage of K.

Theorem 1 Given a plant represented as an NSM G and $B \subseteq L_\omega(G)$, suppose there exists K such that $B = lim(K)$. Then $supC^\omega(B) = lim(supCD(K))$.

An algorithm that computes supremal controllable sublanguage can be easily modified to compute $supCD(K)$, which can then be used to compute $supC^\omega(B)$ using the result of the above theorem. The complexity of computation is polynomial when G is deterministic and B is deterministic ω-regular [1]. The following corollary provides a direct test for ω-controllability, the complexity of which is also polynomial under the same assumptions on G and B.

Corollary 1 Under the hypothesis of Theorem 1, B is ω-controllable iff $supD(K)$ is controllable.

Next theorem provides an iterative technique for computing $supC^\omega(B)$ in the general setting.

Theorem 2 Given a plant represented as an NSM G and $B \subseteq L_\omega(G)$, define the following decreasing sequence of ω-languages: $B_0 := B$; $B_{k+1} := lim(supCD(pr(B_k))) \cap L_\omega(G)$. Suppose there exists $n \geq 0$ such that $B_{n+1} = B_n$, then $B_n = supC^\omega(B)$.

References

[1] R. Kumar and V. K. Garg. *Modeling and Control of Logical Discrete Event Systems.* Kluwer Academic Publishers, Boston, MA, 1994. To be published.

[2] R. Kumar, V. K. Garg, and S. I. Marcus. On supervisory control of sequential behaviors. *IEEE Transactions on Automatic Control*, 37(12):1978–1985, December 1992.

[3] P. J. Ramadge. Some tractable supervisory control problems for discrete event systems modeled by buchi automata. *IEEE Transactions on Automatic Control*, 34(1):10–19, 1989.

[4] J. G. Thistle. *Control of infinite behavior of discrete-event systems.* PhD thesis, University of Toronto, Toronto, Canada, January 1991.

[5] S. Young, D. Spanjol, and V. K. Garg. Control of discrete event systems modeled with deterministic Buchi automata. In *Proceedings of 1992 American Control Conference*, pages 2809–2813, Chicago, IL, 1992.

INDEX OF AUTHORS

Aazhang, B. 382
Al-Dhahir, N. 623
Alexander, P. D. 345
Altman, E. 396
Anantharam, V. 235
Arapostathis, A. 160, 525
Azad, F. 794

Bambos, N. 563
Bar-Ness, Y. 362
Barbeau, M. 709
Barmish, B. R. 51
Baysoy, C. 304
Becker, G. 1
Bejczy, A. K. 641
Berger, T. 822
Bernstein, B. 794
Bhattacharya, P. 748
Blaum, M. 200
Bouthellier, P. R. 21
Brandin, B. A. 699, 881
Brandt, R. D. 294
Brandt-Pearce, M. 133

Cao, C. 871
Chan, A. Y. J. 804
Chen, D. Z. 671
Cheng, X. 765
Chiu, C.-T. 93
Chiu, Y.-J. 822
Chong, E. K. P. 167, 719
Choo, Y. 160, 525
Chu, H.-C. 434
Cidon, I. 553
Cioffi, J. 623
Clarkson, P. M. 832
Collins, T. 794
Costello, D. J., Jr. 196
Cruz, R. L. 245
Cutts, K. 424

Dai, L. 728
De Boer, D. F. 113
de Jong, J. 324
de la Sen, M. 41
De Veciana, G. 254
Dindy-Bolongo, S. 673
Doerschuk, P. C. 755, 814
Dolinar, S. 60
Duel-Hallen, A. 335

Ekroot, L. 60
Elta, M. 784

Fain, E. A. 372
Fan, M. K. H. 513
Fan, Q. 832
Fitz, M. P. 103
Floudas, C. A. 51
Freudenberg, J. 784

Garcia-Esteban, M. 186
Garg, V. K. 900
Gass, J. H., Jr. 143
Geiger, D. 738
Gelfand, S. B. 611
Georghiades, C. N. 131, 841
Ghosh, B. K. 21, 651
Gibson, M. A. 765
Greve, D. 765
Guo, C. 641, 651

Han, J. C. 841
Han, Y. S. 93
Hartmann, C. R. P. 93
Haug, J. 424
Heitman, G. K. 533
Helleseth, T. 220
Hiriyannaiah, M. 851
Hollot, C. V. 51
Holloway, L. E. 851
Horn, G. B. 75
Hsu, I. 264
Huang, H.-C. 633
Hui, S. 31
Humes, A. F. P. De C. 414
Humes, C., Jr. 414

Jang, H. S. 314

Kan, W. Y. 755
Kashyap, R. L. 176
Kawas Kaleh, G. 483
Kesidis, G. 254
Khayrallah, A. S. 503
Kiely, A. 60
Kinney, L. L. 304
Klapper, A. 464
Knight, T. J. 765
Konstantopoulos, T. 235
Krogh, B. H. 765, 890
Krogmeier, J. V. 611, 755
Kschischang, F. R. 75
Kulkarni, S. R. 167
Kumar, P. V. 220
Kumar, R. 861, 900
Kumaran, K. 738
Kuo, W. 103
Kuschewski, J. G. 31

Lehnert, J. S. 434, 454
Levi, R. 274
Li, Q. 124, 230
Lin, F. 294, 871
Lin, W. 60
Litva, J. 804
Liu, L. 603, 617
Longman, R. W. 314, 324
Looze, D. P. 794

Mackenthun, K., Jr. 493
Madyastha, R. K. 382
Makungu, M. 709
Malhotra, S. 176
Maloney, L. T. 738
Martinez-Guerra, R. 523
McEliece, R. J. 60
McWilliams, L. H. 11
Michel, A. N. 284
Miller, G. 603, 617
Min, P. S. 583
Mohan, C. K. 93
Mountford, T. S. 563

Nassar, C. R. 121
Niinomi, T. 890
Noneaker, D. L. 143

Paarmann, L. D. 633
Packard, A. 1
Palmer, E. 774
Park, Y. 719
Perez, L. C. 196
Perkins, J. R. 404
Peterson, R. 424
Pieris, G. 593
Poolla, K. 774
Prabhakar, B. 563
Pursley, M. B. 143

Rashap, B. 784
Rasmussen, L. K. 345
Ren, W. 774
Rimoldi, B. 124, 230
Rom, R. 553
Ryu, Y. S. 324

Sadowsky, J. S. 473
Saidi, H. 583
Sain, M. K. 11
Sarvis, J. P. 210
Sasaki, G. H. 593
Satish, A. 176
Sawan, E. 673
Schlegel, C. 355
Shanbhag, A. G. 200
Shavitt, Y. 553
Shayman, M. A. 861
Shen, B.-Z. 85

Shi, L. 683
Shwartz, A. 274
Siala, M. 483
Siegel, P. H. 200
Sincerbox, G. T. 200
Sira-Ramirez, H. 186
Siveski, Z. 362
Skawratanond, C. 593
Solcz, E. J. 324
Soleymani, M. R. 121
Spanos, C. 774
Srikant, R. 404
St-Denis, R. 709
Stark, W. E. 454
Szczerba, R. J. 671

Tahernezhadi, M. 603, 617
Tarn, T. J. 641, 651, 661
Tassiulas, L. 392
Tempo, R. 51
Trott, M. D. 210
Tsai, J. T. 245
Tzeng, K. K. 85

Uhran, J. J., Jr. 671
Urbanke, R. 230

Varanasi, M. K. 372
Vardy, A. 200
Varvarigos, E. 543
Velidi, R. 131
Viehofer, F. 362

Walrand, J. 264
Wang, I-J. 167
Wang, K. 284
Wang, W.-J. 513
Waxman, B. M. 573
Wei, L. 355
Wicker, S. B. 444
Wu, F. 1
Wu, H.-Y. 335
Wu, Y. 661

Xi, N. 661

Yang, X. H. 1
Yaz, E. E. 153
Ye, H. 284
Yu, Z. 651

Zak, S. H. 31
Zazanis, M. 691
Zhu, W. 748
Ziemer, R. E. 113

UNIVERSITY OF ILLINOIS-URBANA
3 0112 033279750